

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日                      2 0 0 3 年    4 月 2 5 日  
Date of Application:

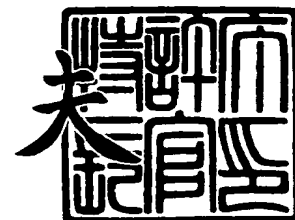
出 願 番 号                      特 願 2 0 0 3 - 1 2 2 2 9 2  
Application Number:  
[ST. 10/C]:                      [ J P 2 0 0 3 - 1 2 2 2 9 2 ]

出      願      人                      インターナショナル・ビジネス・マシーンズ・コーポレーシ  
Applicant(s):                      ョン

2 0 0 3 年    7 月 2 2 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 JP9030035

【提出日】 平成15年 4月25日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 17/30

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ビー・エム株式会社 東京基礎研究所内

【氏名】 今村 剛

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ビー・エム株式会社 東京基礎研究所内

【氏名】 古関 聡

【特許出願人】

【識別番号】 390009531

【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】 100086243

【弁理士】

【氏名又は名称】 坂口 博

【代理人】

【識別番号】 100091568

【弁理士】

【氏名又は名称】 市位 嘉宏

【代理人】

【識別番号】 100108501

【弁理士】

【氏名又は名称】 上野 剛史

## 【復代理人】

【識別番号】 100104880

【弁理士】

【氏名又は名称】 古部 次郎

## 【選任した復代理人】

【識別番号】 100118201

【弁理士】

【氏名又は名称】 千田 武

## 【手数料の表示】

【予納台帳番号】 081504

【納付金額】 21,000円

## 【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9706050

【包括委任状番号】 9704733

【包括委任状番号】 0207860

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 X P a t h 評価方法、これを用いた X P a t h 評価装置及び情報処理装置

【特許請求の範囲】

【請求項 1】 コンピュータを用いて XML 文書に対する X P a t h の評価を行う X P a t h 評価方法であって、

処理対象の前記 XML 文書を構成する XML イベント列を順次入力する第 1 のステップと、

入力された個々の XML イベントに対して X P a t h を順次評価すると共に、所定の XML イベントに対して当該 X P a t h が部分的に成立した場合に、当該 X P a t h の部分的な評価の結果に関する情報を所定の記憶手段に保持する第 2 のステップと、

前記 XML イベント列の入力に伴って、前記記憶手段に保持されている部分的な評価の結果を考慮して前記 X P a t h の部分的な評価を繰り返し、前記 X P a t h の最後の部分が成立した場合に、前記 XML 文書に対して当該 X P a t h が成立したと評価する第 3 のステップと

を含むことを特徴とする X P a t h 評価方法。

【請求項 2】 前記第 2 のステップは、

評価対象である X P a t h を表現するオートマトンを生成するステップと、

入力された個々の XML イベントに基づいて、前記オートマトンの状態を遷移させることにより、前記 X P a t h の部分的な評価を行うと共に、当該オートマトンの状態として当該部分的な評価の結果を保持するステップと

を含むことを特徴とする請求項 1 に記載の X P a t h 評価方法。

【請求項 3】 前記第 2 のステップは、

評価対象である X P a t h をスタック要素の列で表現した第 1 のスタックを生成するステップと、

入力された個々の XML イベントに基づいて前記処理対象の XML 文書の入れ子構造を解析する第 2 のスタックを生成し、前記第 1 のスタックと当該第 2 のスタックとを比較することにより、前記 X P a t h の部分的な評価を行うステップ

と

を含むことを特徴とする請求項 1 に記載の X P a t h 評価方法。

【請求項 4】 前記第 2 のステップは、

入力された個々の XML イベントの入力に基づいて前記処理対象の XML 文書の文書構造を示す文書木を順次生成するステップと、

前記文書木の生成に伴い、生成された部分までの当該文書木を用いて X P a t h の評価を行うステップと

を含むことを特徴とする請求項 1 に記載の X P a t h 評価方法。

【請求項 5】 XML 文書を構成する XML イベント列を入力し、個々の XML イベントに対して X P a t h を順次評価すると共に、所定の XML イベントに対して当該 X P a t h が部分的に成立した場合に、当該 X P a t h の部分的な評価の結果に関する情報を保持していき、当該 X P a t h の最後のステップが成立した場合に、当該 XML 文書に対して当該 X P a t h が成立したと評価する評価実行部と、

処理対象の XML 文書を構成する XML イベント列を入力し、前記評価実行部に順次転送する XML イベント転送部と

を備えることを特徴とする X P a t h 評価装置。

【請求項 6】 評価対象である X P a t h を表現するオートマトンを生成するオートマトン生成部をさらに備え、

前記評価実行部は、前記 XML イベント転送部から転送された XML イベントに基づいて、前記オートマトン生成部にて生成された前記オートマトンの状態を遷移させることにより、前記 X P a t h の部分的な評価を行うと共に、当該オートマトンの状態として当該部分的な評価の結果を保持することを特徴とする請求項 5 に記載の X P a t h 評価装置。

【請求項 7】 評価対象である X P a t h をスタック要素の列で表現した第 1 のスタックを生成するスタック生成部をさらに備え、

前記評価実行部は、前記 XML イベント転送部から転送された XML イベントに基づいて前記処理対象の XML 文書の入れ子構造を解析する第 2 のスタックを生成し、前記スタック生成部にて生成された第 1 のスタックと当該第 2 のスタック

クとを比較することにより、前記X P a t hの部分的な評価を行うことを特徴とする請求項5に記載のX P a t h評価装置。

【請求項8】 XML文書を構成するXMLイベント列を入力し、個々のXMLイベントの入力に伴い、入力された当該XMLイベントに基づいて当該XML文書の文書構造を示す文書木を順次生成する文書木生成部と、

処理対象のXML文書を構成するXMLイベント列を入力し、前記文書木生成部に順次転送するXMLイベント転送部と、

前記文書木生成部による前記文書木の生成に伴い、生成された部分までの当該文書木を用いてX P a t hの評価を行う評価実行部とを備えることを特徴とするX P a t h評価装置。

【請求項9】 前記評価実行部は、前記文書木を用いたX P a t hの評価において、当該X P a t hが部分的に成立した場合に、当該X P a t hの部分的な評価の結果に関する情報を保持することを特徴とする請求項8に記載のX P a t h評価装置。

【請求項10】 処理対象のXML文書を解析してXMLイベント列を生成するXMLパーサと、

前記XMLパーサにて生成された前記XMLイベント列を順次入力し、入力した個々のXMLイベントに対してストリーミング処理にてX P a t hの評価を行うX P a t h評価部と、

前記XMLパーサで生成された前記XMLイベントを入力し、前記X P a t h評価部によるX P a t hの評価結果に応じて、当該XMLイベントにて構成される前記XML文書に対する処理を実行するアプリケーション実行部とを備え、

前記X P a t h評価部は、前記個々のXMLイベントに対してX P a t hを順次評価すると共に、所定のXMLイベントに対して当該X P a t hが部分的に成立した場合に、当該X P a t hの部分的な評価の結果に関する情報を保持していき、当該X P a t hの最後のステップが成立した場合に、前記XML文書に対して当該X P a t hが成立したと評価することを特徴とする情報処理装置。

【請求項11】 前記X P a t h評価部は、評価対象であるX P a t hを表現するオートマトンを生成し、前記XMLパーサにて生成された前記XMLイベ

ントに基づいて、当該オートマトンの状態を遷移させることにより、前記X P a t hの部分的な評価を行うと共に、当該オートマトンの状態として当該部分的な評価の結果を保持することを特徴とする請求項10に記載の情報処理装置。

【請求項12】 前記X P a t h評価部は、評価対象であるX P a t hをスタック要素の列で表現した第1のスタックを生成し、前記XMLパーサにて生成された前記XMLイベントに基づいて前記処理対象のXML文書の入れ子構造を解析する第2のスタックを生成し、当該第1のスタックと当該第2のスタックとを比較することにより、前記X P a t hの部分的な評価を行うことを特徴とする請求項10に記載の情報処理装置。

【請求項13】 前記X P a t h評価部は、前記XMLパーサにて生成された個々の前記XMLイベントの入力に伴い、入力された当該XMLイベントに基づいて前記処理対象のXML文書の文書構造を示す文書木を順次生成し、生成された部分までの当該文書木を用いてX P a t hの評価を行うことを特徴とする請求項10に記載の情報処理装置。

【請求項14】 コンピュータを制御して、XML文書に対するX P a t hの評価を行うプログラムであって、

処理対象の前記XML文書を構成するXMLイベント列を順次入力する第1の処理と、

入力された個々のXMLイベントに対してX P a t hを順次評価すると共に、所定のXMLイベントに対して当該X P a t hが部分的に成立した場合に、当該X P a t hの部分的な評価の結果に関する情報を所定の記憶手段に保持する第2の処理と、

前記XMLイベント列の入力に伴って、前記記憶手段に保持されている部分的な評価の結果を考慮して前記X P a t hの部分的な評価を繰り返し、前記X P a t hの最後の部分が成立した場合に、当該X P a t hが成立したと評価する第3の処理と

を前記コンピュータに実行させることを特徴とするプログラム。

【請求項15】 前記第2の処理では、  
評価対象であるX P a t hを表現するオートマトンを生成する処理と、

入力された個々のXMLイベントに基づいて、前記オートマトンの状態を遷移させることにより、前記XPathの部分的な評価を行うと共に、当該オートマトンの状態として当該部分的な評価の結果を保持する処理とを前記コンピュータに実行させることを特徴とする請求項14に記載のプログラム。

【請求項16】 前記第2の処理では、

評価対象であるXPathをスタック要素の列で表現した第1のスタックを生成する処理と、

入力された個々のXMLイベントに基づいて前記処理対象のXML文書の入れ子構造を解析する第2のスタックを生成し、前記第1のスタックと当該第2のスタックとを比較することにより、前記XPathの部分的な評価を行う処理とを前記コンピュータに実行させることを特徴とする請求項14に記載のプログラム。

【請求項17】 前記第2の処理では、

入力された個々のXMLイベントの入力に基づいて前記処理対象のXML文書の文書構造を示す文書木を順次生成する処理と、

前記文書木の生成に伴い、生成された部分までの当該文書木を用いてXPathの評価を行う処理とを前記コンピュータに実行させることを特徴とする請求項14に記載のプログラム。

【請求項18】 請求項14から請求項17のいずれかに記載のプログラムを、コンピュータが読み取り可能に記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、XML (Extensible Markup Language) 文書の解析技術に関し、特にXPath (XML Path Language) の評価を行う解析システム及びその解析方法に関する。

【0002】



**【従来の技術】**

今日、インターネットに代表されるコンピュータネットワークのパケット交換に用いられる文書記述言語として、文書の論理構造や構成要素の意味を自由に定義できるXMLが広く普及している。XML文書をアプリケーションで利用する際に、処理対象であるXML文書を解析することが必要となるが、このときXPathが利用される。XPathは、XML文書の特定の部分を指し示す構文を規定する言語であり、W3C (World Wide Web Consortium) 勧告の仕様である。XPathを利用することにより、所定のXML文書が特定の論理構造や構成要素を持つかどうかを解析することができる（例えば、非特許文献1参照）。

**【0003】**

また、XML文書の構造は、木構造で表現することができ、W3C勧告の仕様であるDOM (Document Object Model) 等を用いて表現される。XPathを用いてXML文書を解析する場合、通常、DOM等で表現されたXML文書全体を対象として走査し、XPathにて記述された構造を当該XML文書が持つかどうかを調べる。したがって、XML文書の解析システムは、解析対象であるXML文書全体を所定のメモリに読み込んで処理を行う必要があった。

**【0004】****【非特許文献1】**

“XML Path Language (XPath) Version 1.0”、[online]、1999年11月16日、[2003年3月4日検索]、インターネット<<http://www.w3.org/TR/xpath>>

**【0005】****【発明が解決しようとする課題】**

上記のように、XPathを用いてXML文書の解析を行う場合、従来は、解析対象であるXML文書全体を、一旦メモリの作業領域に読み込む必要があった。しかし、解析対象であるXML文書のデータサイズが巨大である場合、これをDOM等の記述として読み込むのは、多大なメモリ使用量を要し、またDOMを作成、操作する処理に多大な時間を要するという問題があった。

**【0006】**

また、SAX (The Simple API for XML) やXNIのようなストリーミングベースのAPI (Application Program Interface) を用いる場合、APIはストリーミング形式で取得されるXML文書を順次処理できるにも関わらず、XPathによる解析を行うために、一旦XML文書全体を読み込んでからでなければ処理を開始できないのは効率が悪い。

#### 【0007】

そこで本発明は、上記の問題に鑑み、XML文書をストリーミング処理しながらXPathの評価を行う解析システム及びその解析方法を実現することを目的とする。

#### 【0008】

##### 【課題を解決するための手段】

上記の目的を達成する本発明は、コンピュータを用いてXML文書に対するXPathの評価を行う、次のようなXPath評価方法として実現される。このXPath評価方法は、処理対象のXML文書を構成するXMLイベント列を順次入力する第1のステップと、入力された個々のXMLイベントに対してXPathを順次評価すると共に、所定のXMLイベントに対してこのXPathが部分的に成立した場合に、このXPathの部分的な評価の結果に関する情報を所定の記憶手段に保持する第2のステップと、XMLイベント列の入力に伴って、記憶手段に保持されている部分的な評価の結果を考慮してXPathの部分的な評価を繰り返し、XPathの最後の部分が成立した場合に、処理対象であるXML文書に対して、このXPathが成立したと評価する第3のステップとを含むことを特徴とする。

このように、入力したXMLイベントに対して個別にXPathの評価を行っていき、その部分的な評価結果を蓄積しながら、XPath全体が成立するかどうかを判断することにより、ストリーミング処理によるXPathの評価が可能となる。

#### 【0009】

この第2のステップで行われるXPathの評価及び部分的な評価の結果を保持する具体的な手法として、本発明では、オートマトンを用いた手法、スタック

を用いた手法、XML イベント列の入力に伴って順次生成される文書木を用いた手法を提示する。

オートマトンを用いた手法では、評価対象である XPath を表現するオートマトンを生成し、個々の XML イベントに基づいて、このオートマトンの状態を遷移させることにより、XPath の評価を行う。この部分的な評価の結果は、オートマトンの状態として保持される。

#### 【0010】

スタックを用いた手法では、評価対象である XPath をスタック要素の列で表現した第1のスタックを生成し、個々の XML イベントに基づいて処理対象の XML 文書の入れ子構造を解析する第2のスタックを生成する。そして、第1のスタックと第2のスタックとを比較することにより、XPath の評価を行う。この部分的な評価の結果は、第2のスタックのスタック要素として保持される。

#### 【0011】

XML イベント列の入力に伴って順次生成される文書木を用いた手法では、入力された個々の XML イベントの入力に基づいて処理対象の XML 文書の文書構造を示す文書木を順次生成していき、この文書木の生成に伴い、生成された部分までの文書木を用いて XPath の評価を行う。この場合、XML イベント列の入力に伴って文書木が作成されていくので、文書木において新たなノードが追加されるたびに XPath を評価していき、XPath が成立するかどうかを調べれば良い。しかし、XPath が成立するノードが複数存在する場合もあるので、部分的な評価の結果の情報を保持しておき、後続の XML イベントに対する評価の際に用いることができる。

#### 【0012】

また、上記の目的を達成する他の本発明は、次のように構成された XPath 評価装置としても実現される。この XPath 評価装置は、ストリーミング処理による XPath の評価を実行する評価実行部と、処理対象の XML 文書を構成する XML イベント列を入力し、評価実行部に順次転送する XML イベント転送部とを備える。この評価実行部は、XML イベント転送部から転送された個々の XML イベントに対して XPath を順次評価すると共に、所定の XML イベント

トに対して、このX P a t hが部分的に成立した場合に、このX P a t hの部分的な評価の結果に関する情報を保持していく。そして、このX P a t hの最後のステップが成立した場合に、XML文書に対して、このX P a t hが成立したと評価することを特徴とする。

#### 【0013】

このX P a t h評価装置は、上述したX P a t hの評価及び部分的な評価の結果を保持する具体的な手法として、オートマトンを用いた手法を実現するため、X P a t hを表現するオートマトンを生成するオートマトン生成部をさらに備える構成とすることができる。同様に、スタックを用いた手法を実現するため、X P a t hをスタック要素の列で表現した第1のスタックを生成するスタック生成部をさらに備える構成とすることができる。

#### 【0014】

また、本発明の他のX P a t h評価装置は、XML文書を構成するXMLイベント列を入力し、個々のXMLイベントの入力に伴い、入力されたXMLイベントに基づいてXML文書の文書構造を示す文書木を順次生成する文書木生成部と、処理対象のXML文書を構成するXMLイベント列を入力し、この文書木生成部に順次転送するXMLイベント転送部と、この文書木生成部による文書木の生成に伴い、生成された部分までの文書木を用いてX P a t hの評価を行う評価実行部とを備えることを特徴とする。

#### 【0015】

さらに本発明は、XMLパーサと、上述したX P a t h評価装置の機能を備えたX P a t h評価部と、XMLパーサで生成されたXMLイベントを入力し、X P a t h評価部によるX P a t hの評価結果に応じて、入力したXMLイベントにて構成されるXML文書に対する処理を実行するアプリケーション実行部とを備える情報処理装置として実現することもできる。

#### 【0016】

また本発明は、コンピュータを制御して上述したX P a t h方法の各ステップに対応する処理を実行させるプログラム、またはコンピュータを上記X P a t h評価装置あるいは情報処理装置として機能させるプログラムとしても実現される

。このプログラムは、磁気ディスクや光ディスク、半導体メモリ、その他の記録媒体に格納して配布したり、ネットワークを介して配信したりすることにより、提供することができる。

#### 【0017】

##### 【発明の実施の形態】

以下、添付図面に示す実施の形態に基づいて、この発明を詳細に説明する。

本発明は、ストリーミング形式で入力される処理対象のXML文書に対して、当該XML文書の入力に応じて順次XPathの評価を行う。このストリーミング処理によるXPathの評価を実現する実施の形態として、以下ではオートマトンを用いた実施の形態（実施の形態1）、スタックを用いた実施の形態（実施の形態2）、ツリーを用いた実施の形態（実施の形態3）を説明する。なお、これらの実施の形態は、パーソナルコンピュータやワークステーション、その他のコンピュータ装置、PDA（Personal Digital Assistant）、携帯電話等の種々の情報処理装置にて実現される。

#### 【0018】

##### 〔実施の形態1〕

実施の形態1では、オートマトンを用いることにより、XPathの評価のストリーミング処理を実現する。

図1は、本実施の形態を実現するのに好適な情報処理装置のハードウェア構成の例を模式的に示した図である。

図1に示す情報処理装置は、演算手段であるCPU（Central Processing Unit：中央処理装置）101と、M/B（マザーボード）チップセット102及びCPUバスを介してCPU101に接続されたメインメモリ103と、同じくM/Bチップセット102及びAGP（Accelerated Graphics Port）を介してCPU101に接続されたビデオカード104と、PCI（Peripheral Component Interconnect）バスを介してM/Bチップセット102に接続されたハードディスク105、ネットワークインターフェイス106及びUSBポート107と、さらにこのPCIバスからブリッジ回路108及びISA（Industry Standard Architecture）バスなどの低速なバスを介してM/Bチップセット102に接

続されたフロッピーディスクドライブ109及びキーボード/マウス110とを備える。

なお、図1は本実施の形態を実現するコンピュータのハードウェア構成を例示するに過ぎず、本実施の形態を適用可能であれば、他の種々の構成を取ることができる。例えば、ビデオカード104を設ける代わりに、ビデオメモリのみを搭載し、CPU101にてイメージデータを処理する構成としても良いし、ATA (AT Attachment) などのインターフェイスを介してCD-ROM (Compact Disc Read Only Memory) やDVD-ROM (Digital Versatile Disc Read Only Memory) のドライブを設けても良い。

#### 【0019】

図2は、本実施の形態におけるXPath評価手段（解析システム）の実装例を示す図である。

図2に示すように、本実施の形態による情報処理装置は、処理対象であるXML文書の構文解析（パース）を行うXMLパーサ10と、構文解析されたXML文書に対してXPathの評価を行うXPath評価部20と、XPathの評価を経たXML文書を用いて所定の情報処理を実行するアプリケーション実行部30とを備える。

図2に示すXMLパーサ10、XPath評価部20及びアプリケーション実行部30は、例えば、図1に示したメインメモリ103に展開されたプログラムにてCPU101を制御することにより実現される仮想的なソフトウェアブロックである。

#### 【0020】

図2において、XMLパーサ10は、処理対象のXML文書を入力して構文解析を行い、解析結果を通知するXMLイベントを出力してXPath評価部20に送る。XMLイベントは、解析対象のXML文書の木構造（ドキュメントツリー）におけるノードごとに順次、XPath評価部20へ送られる。XMLパーサ10としては、XML文書を扱う情報処理装置で従来から用いられている既存のXMLパーサを用いることができる。なお、本実施の形態において、入力対象となるXML文書は整形形式であることを前提とする。

**【0021】**

X P a t h 評価部 20 は、XML パーサ 10 から XML イベントを受け取り、予め与えられた X P a t h の評価を行う。

図 3 は、X P a t h 評価部 20 の機能構成を説明する図である。

図 3 に示すように、本実施の形態の X P a t h 評価部 20 は、オートマトン生成部 21 と、XML イベント転送部 22 と、評価実行部 23 とを備える。

**【0022】**

オートマトン生成部 21 は、予め与えられた X P a t h からオートマトン（状態遷移機械）を生成する。本実施の形態では、SAX や XNI 等のストリーミング形式で実行される API を用いたアプリケーションに対して X P a t h を評価する。したがって、ストリーミング処理を行うという制約上、評価対象となる X P a t h の機能は一部制限される。例えば、評価において既に通知された XML イベントを参照することが必要となる基準点（axis）や述語（predicate）は実現されない。しかし、実際の多くのアプリケーションにおける処理では、かかる制限が問題となることはない。

**【0023】**

ここで、本実施の形態における X P a t h の制限を具体的に提示する。

1. X P a t h は、ロケーションパスとする。ただし、絶対ロケーションパスは、対象である XML 文書の木構造のルートノードから X P a t h を評価するときに限る。
2. 基準点は、フォワード基準点のみとする。すなわち、次のいずれかとする。

s e l f  
c h i l d  
d e s c e n d a n t  
d e s c e n d a n t - o r - s e l f  
f o l l o w i n g  
f o l l o w i n g - s i b l i n g  
a t t r i b u t e  
n a m e s p a c e

ただし、ある条件と特別な実装の上でリバース基準点も使用することができる。  
これについては後述する。

3. 述語については次のように扱う。

(1) 演算子は、次のいずれかとする。

o r  
a n d  
=  
!=

(2) プリミティブは、次のいずれかとする。

・ロケーションパス。ただし、その基準点は次のいずれかとする。

a t t r i b u t e  
n a m e s p a c e  
s e l f  
p a r e n t  
a n c e s t o r  
a n c e s t o r - o r - s e l f

・リテラル

・数

・関数呼び出し。ただし、次のものを除く。

l a s t ( )  
i d ( o b j e c t )

ルートノード及び要素ノードに対する s t r i n g ( o b j e c t ? )

【0024】

さて、X P a t h において、省略記法を用いないロケーションパスは、基準点とノードテストと述語の接続（ステップと呼ばれる）を記号 / で区切って並べたもので表される。オートマトン生成部 21 は、各ステップの評価状態をオートマトンで表現する。このオートマトンは、次のような状態遷移を行う。

図 4 は、オートマトン生成部 21 にて生成されるオートマトンを模式的に示す図である。



まず、各ステップ（個数をNとする）を左（先頭）から順番に番号付けし、所定のトークンが読み込まれた時点でステップKが成立している状態をステートKとする。ここで、ステップKが成立している状態とは、ステップKを満たすノードの開始タグトークンが読み込まれ、その終了タグがまだ読み込まれていない状態を表すものとする。また、初期状態はステート0で表すとする。

今、ステップKを満たすノードの開始タグトークンが読み込まれ、現在の状態がステートKになったとする。このステートKの状態ですテツプK+1を満たすノードが読み込まれた場合は、ステートK+1に状態が移る。一方、ステートKの状態ですテツプKの終了タグが読み込まれた場合は、ステートK-1に状態が移る。このような状態遷移を繰り返し、ステートNに到達すると、ロケーションパスが満たされたと判断される。

オートマトン生成部21によって生成された上記のようなオートマトンは、例えば図1のメインメモリ103の作業領域に保持され、後述する評価実行部23の処理において使用される。

#### 【0025】

XMLイベント転送部22は、XMLパーサ10から出力されたXMLイベントを入力し、XML文書の木構造におけるノードごとに評価実行部23へ順次転送する。また、当該XMLイベントをアプリケーション実行部30へも順次転送する。

#### 【0026】

評価実行部23は、オートマトン生成部21にて生成されたオートマトンを取得し、XMLイベント転送部22から転送されたXMLイベントを受け取って、XML文書のトークン列を順次読み込みながらオートマトンにおける状態間の遷移を行う。そして、ロケーションパスを満たすノードに対応するトークンが読み込まれた時点でそのロケーションパスが満たされたと評価する。この評価結果は、評価イベントとしてアプリケーション実行部30に送られる。評価実行部23による具体的な評価処理の手順については後述する。

#### 【0027】

図2に示したアプリケーション実行部30は、XPath評価部20のXML

イベント転送部 22 から転送される XML イベントと評価実行部 23 から出力される評価イベントとを入力し、ストリーミング処理にて XML 文書に対する処理を実行する。

#### 【0028】

次に、本実施の形態の XPath 評価部 20 による処理について、さらに詳細に説明する。

上述したように、本実施の形態では、まず評価対象である XPath を表現するオートマトンを生成し保持しておく。そして、処理対象の XML 文書を構成する XML イベント列を順次入力し、この XML イベントに基づいて、オートマトンの状態を遷移させることにより、XPath の部分的な評価を行う。XPath の部分的な評価の結果は、オートマトンの状態として保持される。

ここでは簡単のために、ロケーションパスは、述語が無いものとし、基準点では child と descendant のみを考える。following-sibling、following、self、descendant-or-self、attribute、namespace を基準点とするステップや述語が存在するステップの取り扱いについては後述する。

#### 【0029】

図 5 は、本実施の形態による処理に用いられるロケーションパス、オートマトン及び処理対象の XML 文書の例を示す図である。

図 5 (A) に示すロケーションパスが与えられ、このロケーションパスの各ステップを左からステップ 1、ステップ 2、ステップ 3 と呼ぶことにする。オートマトン生成部 21 により、このロケーションパスから生成されるオートマトンの状態は、図 5 (B) に示すようになる。

今、図 5 (C) に示す XML 文書が入力されたとする。この XML 文書は、XML パーサ 10 にて構文解析され、ノードごとの XML イベントとして XPath 評価部 20 に送られ、XML イベント転送部 22 にて評価実行部 23 に順次転送される。なお、図 5 (C) において、<title></title>のように要素内容がない場合は<title/>と省略表記がなされている（以下、他の XML 文書についても同様）。

**【0030】**

評価実行部23は、オートマトン生成部21にて生成された上記のオートマトンを用い、XMLイベント転送部22から転送されたXMLイベントに基づいて当該オートマトンの状態遷移を行って、当該オートマトンに対応するロケーションパスの評価を行う。なお、実際の処理において、オートマトンの状態の遷移及び評価は、XML文書の入れ子構造を解析するスタック（XMLイベント列用のスタック）を、例えば図1のメインメモリ103のメモリ空間に構築して行う。すなわち、評価実行部23は、XMLパーサ10から送られるXMLイベント列を、このXMLイベント列用スタックに順次保持していく。

このXMLイベント列用スタックは、開始タグが読み込まれたことを条件に、そのタグに対応するノードをプッシュし、終了タグが読み込まれたことを条件に、当該ノードをポップする。また、スタックのボトム(bottom)はルートノードを表すものとする。

**【003.1】**

したがって、評価実行部23は、XMLイベント列の入力に応じて、XMLイベント列用スタックへのプッシュとポップを順次行っていく。この操作は、ロケーションパスが相対であるか絶対であるかに関わらず、XML文書の開始から終了まで一貫して行われる。この操作中、所定の開始タグが読み込まれ、そのタグに対応するノードがXMLイベント列用スタックにプッシュされた場合はいつでも、当該ノードが（現在の状態をステートKとして）ステートK+1に対応するステップ（すなわちステップK+1）を満たしているかどうかを判定する。この判定結果が真であった場合、現在の状態をステートK+1とする。このとき、スタック要素とステートの対応付けをしておく。つまり、プッシュされたノードとそのノードによって成立したステートとのマッピングを構成する。ここで、ステート0に対しては、ロケーションパスが相対パスである場合にはコンテキストノードに対応したスタック要素が対応付けられる。またロケーションパスが絶対パスである場合はスタックのボトム（すなわちルートノード）が対応付けられるものとする。一方、XMLイベント列用スタックからのポップが行われた場合、ポップされたエレメントがステートに対応付けされているかどうかを検出し、対応

付けされていたならば、状態を1つ前に戻す。

#### 【0032】

以下、図5を参照し、順を追って説明する。

まず、初期状態は状態0であり、トークン<document>が読み込まれ、ノードdocumentがプッシュされる。このときのXMLイベント列用スタックの状態は、

bottom | document

となる。このノードdocumentは、ステップ1を満たすので、状態0から状態1へ状態遷移が行われる。

#### 【0033】

次に、トークン<title>が読み込まれ、ノードtitleがプッシュされる。このときのXMLイベント列用スタックの状態は、

bottom | document | title

となる。このノードtitleは、状態1に対応付けられたノードdocumentのchild::chapterという条件を満たさない。すなわち、ステップ2を満たさないので、状態遷移は行われず、そのままトークンの読み込みが続けられる。

#### 【0034】

ノードtitleの終了タグであるトークン</title>が読み込まれてノードtitleがポップされた後、続いてトークン<chapter>が読み込まれ、ノードchapterがプッシュされる。このノードはステップ2を満たすので、状態1から状態2へ状態遷移が行われる。このときのXMLイベント列用スタックは、

bottom | document | chapter

となる。次に、トークン</chapter>が読み込まれ、ノードchapterのポップが行われる。このとき、ポップされるスタック要素には、ステート2が対応付けられている。したがってこの場合は、ポップと共に、オートマトンの状態が、ステート2からその1つ前の状態であるステート1へ遷移される。

#### 【0035】

この後、再びトークン<chapter>が読み込まれ、ノードchapterがプッシュされて、ステート1からステート2へ状態遷移が行われる。そしてさらに、トークン<section>が読み込まれ、ノードsectionがプッシュされると、このノードはステップ3を満たすので、ステート2からステート3へ状態遷移が行われる。この時点で評価実行部23は、ロケーションパスが成立したと判断し、評価結果を示す評価イベントをアプリケーション実行部30へ送信する。

なお、以上の説明では、評価実行部23はロケーションパスが成立した場合に評価イベントを出力することとしたが、XMLイベントを1つ処理する度に成立または不成立を示す評価イベントを出力するようにしても良い。

#### 【0036】

次に、他のフォワード基準点(axis)の取り扱いについて説明する。

##### 1. attribute、namespaceの取り扱い。

基準点がattributeであるステップが含まれるロケーションパスを評価するには、状態遷移した際に、遷移後の状態の1つ先の状態に対応するステップを検出する。そして、そのステップがattributeであれば、最後に入力されたトークンの属性を用いて、当該ステップの評価を行う。この評価でノードがステップを満足した場合は、状態を1つ進める。一方、ノードがステップを満足しなかった場合は、元々の状態遷移をキャンセルする。

namespaceに関しては、現在のコンテキストにおけるnamespaceを管理しておき、ステップの評価をattributeの場合と同様に実行する。

なお、attribute及びnamespaceに関しては、これらが成立

する過程では、XML イベント列用スタックの操作は行われない。したがって、それらの状態と対応付けられるノードは何もないことを表すNULLノードとしておく。

#### 【0037】

##### 2. selfの取り扱い。

基準点がselfであるステップが含まれるロケーションパスを評価するには、状態遷移した際に、遷移後の状態の1つ先の状態に対応するステップを検出する。そして、そのステップがselfであれば、当該ステップの評価を行う。この評価でノードがステップを満足した場合は、状態を1つ進め、その先の状態に対して同じ操作を繰り返す。一方、ノードがステップを満足しなかった場合は、元々の状態遷移をキャンセルする。selfが成立する過程では、XML イベント列用スタックの操作は行われない。したがって、それらの状態と対応付けられるノードは何もないことを表すNULLノードとしておく。

#### 【0038】

##### 3. descendant-or-selfの取り扱い。

基準点がdescendant-or-selfであるステップが含まれるロケーションパスを評価するには、まず、そのステップの基準点がselfであるとみなして、上記2. の操作を行う。この操作においてステップを満足しないと判断された場合、その基準点がdescendantであるとみなして評価を続行する。

#### 【0039】

##### 4. following-siblingの取り扱い。

基準点がfollowing-siblingであるステップが含まれるロケーションパスを評価するには、以下の操作を行う。

まず、ある基準点がfollowing-siblingであるステップをステップKとし、それに対応した状態をステートKとする。基本的には、ステップK-1の成立時にプッシュされたノードがポップされたときのスタックトップノードに対し、そのノードに続いてプッシュされるノードがステップKで選択すべきノードとなる。ただし、ここで注意しなければならないのは、このステップ

Kを満足したノードは、ステップKで選択されるノードの候補であると同時にステップK-1で選択されるノードの候補でもあるということである。例えば、ロケーションパスが、

```
/child::document/child::chapter/following-sibling::chapter
```

であった場合、以下の2番目のchapterは、2番目のステップと、3番目のステップの両方で選択される可能性がある。

```
<document>  
  <chapter>--1  
  </chapter>  
  <chapter>--2  
  </chapter>  
  <chapter>--3  
  </chapter>  
</document>
```

このような両義性は、positionに依存したノード選択を行うときは厳密な解釈が必要である。例えば、以下のロケーションパス

```
/child::document/child::chapter/following-sibling::chapter[1]
```

では、2番目と3番目のchapterを共に選択しなければならない。また、以下のロケーションパス

```
/child::document/child::chapter[2]/following-sibling::chapter
```

では、3番目のchapterを選択しなければならない。

## 【0040】

そこで、本実施の形態では、ノードがステップK-1を満足すると判断された時点でプロセスを2つに分割し、それぞれのプロセスが2通りのロケーションパスの解釈に対応した状態遷移を続行するように処理を行う。1つのプロセスは、この時点で当該ノードと対応付けられた状態の1つ前の状態に遷移し（通常はステートK-2、selfが含まれる場合はさらにその前の状態）、以降の評価を続行する。

もう1つのプロセスでは、次の操作を行う。まず、状態はステートK-1のままとする。ステートK-1の状態において、ノードがステップK-1を満足すると判断されてプッシュされたノードがポップされたときのスタックトップノードがスタックトップである間に何らかのノードがプッシュされた場合は、当該ノードがステップKステップKを満足するかどうかを判定する。満足するならば、状態をステートKに遷移し、プッシュされたノードとステートKとを対応付けて、これ以降の評価を続行する。また、ステートKに対応付けられたノードがポップされたときは、状態をステートK-1に戻す。ステートK-1の状態において、ステップK-1を満足したノードの終了タグトークンが入力された場合、すなわちノードがステップK-1を満足すると判断されてプッシュされたノードがポップされたときのスタックトップノードがポップされた場合は、このプロセスにおいてステップKで選択されるノードがこれ以上存在しないので操作を終了する。

## 【0041】

5. followingの取り扱い。

基準点がfollowingであるステップが含まれるロケーションパスを評価するには、以下の操作を行う。

この場合もロケーションパスに曖昧さが生じる可能性があるため、following-siblingの場合と同じ様なプロセスの分割が必要となる。まず、ある基準点がfollowingであるステップをステップKとし、それに対応した状態をステートKとする。上記4.の処理と同様に、ノードがステップK-1を満足すると判断された時点でプロセスを2つに分割する。1つのプロセスは、この時点で当該ノードと対応付けられた状態の1つ前の状態に遷移し、以降



の評価を続行する。

もう1つのプロセスでは、次の操作を行う。まず、状態はステートK-1のままとする。ステートK-1の状態において、何らかのノードがプッシュされた場合は、ステップKを満足するかどうかを判定する。満足するならば、状態をステートKに遷移し、プッシュされたノードとステートKとを対応付けて、これ以降の評価を続行する。followingの場合は、ステートK-1の状態において、ステップK-1を満足したノードの終了タグトークンが入力されても、状態をステートKのままにする。

#### 【0042】

次に、述語 (predicate) の取り扱いについて説明する。

述語は、基本的には、状態遷移の瞬間にスタックに蓄積された情報から構成できるものについて取り扱うことができる。つまり、ステップKを満たす開始タグトークンが読み込まれた場合、その状態で構成できる当該ノードについての情報、及びスタックに残っている当該ノードの ancestor についての情報を述語に記述することができる。また、各ステートについてカウンタを設けることで、ポジション関数を取り扱うことが可能となる。これまで説明してきたようなステップの成立状態において、そのステップに対応したステートのカウンタを1つ増やす。このカウンタの値に基づいてポジション関数を含む述語を評価し、最終的にノードがステップを満足するか否かを判定する。

#### 【0043】

次に、リバース基準点の取り扱いについて説明する。

本実施の形態では、非常に限定されたロケーションパスにおいて、parent、ancestor、ancestor-or-self といったリバース基準点を使用することができる。上記述語の取り扱いで述べたように、現在のノードの ancestor についての情報はオートマトンの状態として蓄積されている。したがって、ロケーションパスのある地点から右側のステップが全て、parent、ancestor、ancestor-or-self、あるいはselfである場合は、その地点でスタックの操作を一旦停止し、スタックを遡りながら、それらのステップを評価していけば良い。

## 【0044】

図6は、本実施の形態によるXML文書のストリーミング処理を説明するフローチャートである。

図6に示すように、X P a t h評価部20は、初期動作として、オートマトン生成部21により、予め与えられたX P a t hのロケーションパスからオートマトンを生成しておく（S t e p 601）。処理対象であるXML文書が入力され、XMLパーサ10からXMLイベントが送られると（S t e p 602）、X P a t h評価部20の評価実行部23により、上記のようにして、当該XMLイベントに対する先のオートマトンにおける状態遷移が順次判断される（S t e p 603）。この状態遷移が当該オートマトンに対応するロケーションパス（X P a t h）の部分的な評価を意味している。この部分的な評価の結果は、オートマトンの現在の状態として保持される。

そして、このオートマトンの最後の状態に達するまでXMLイベントの入力及び状態遷移の判断を繰り返し、最後の状態に達したならば、このXML文書に対してロケーションパスが成立したと評価され、評価イベントがアプリケーション実行部30へ送られる（S t e p 604、605）。

アプリケーション実行部30は、XMLパーサ10から出力されたXMLイベントと、X P a t h評価部20の評価実行部23から出力された評価イベントとを取得し、当該評価イベントに応じて当該XMLイベントを順次処理していく。

処理対象のXML文書についてXMLパーサ10から送られる全てのXMLイベントに対して、S t e p 602～605の処理を実行した後、本実施の形態によるXML文書のストリーミング処理を終了する（S t e p 606）。

## 【0045】

## 〔実施の形態2〕

実施の形態2では、実施の形態1で説明したオートマトンの状態遷移を用いた処理に相当するX P a t hの評価を、XML文書の入れ子構造を解析するスタックを用いて実現する。すなわち本実施の形態は、X P a t h及びXMLイベント列を共にスタックで表現し、XMLイベントが通知される度にそれらを比較するという方法で実装される。

**【0046】**

本実施の形態は、実施の形態1と同様に、例えば図1に示した情報処理装置にて実現される。

図7は、本実施の形態におけるX P a t h評価手段（解析システム）の実装例を示す図である。

図7に示すように、本実施の形態による情報処理装置は、処理対象であるXML文書の構文解析（パース）を行うXMLパーサ10と、構文解析されたXML文書に対してX P a t hの評価を行うX P a t h評価部40と、X P a t hの評価を経たXML文書を用いて所定の情報処理を実行するアプリケーション実行部30とを備える。

図7に示すXMLパーサ10、X P a t h評価部40及びアプリケーション実行部30は、例えば、図1に示したメインメモリ103に展開されたプログラムにてCPU101を制御することにより実現される仮想的なソフトウェアブロックである。なお、XMLパーサ10及びアプリケーション実行部30は、実施の形態1における対応する構成要素と同様であるため、同一の符号を付して説明を省略する。

**【0047】**

X P a t h評価部40は、XMLパーサ10からXMLイベントを受け取り、予め与えられたX P a t hの評価を行う。

図8は、X P a t h評価部40の機能構成を説明する図である。

図8に示すように、本実施の形態のX P a t h評価部40は、スタック生成部41と、XMLイベント転送部42と、評価実行部43とを備える。

**【0048】**

スタック生成部41は、予め与えられたX P a t hから、当該X P a t hのステップを表現するデータ構造であるスタックを生成する。X P a t hからのスタックの生成は、次のようにして行われる。

すなわち、スタック生成部41は、X P a t hの各ステップに対して、ノードテストと述語とをスタック要素で表現し、基準点に応じてスタックにプッシュする。例えば、c h i l d : : p a r aに対してはp a r aをスタック要素で表現

し、既存のスタックにプッシュする。ここで、ノードテストと述語とは、XML イベント列に対してそれらを適切に評価できる限り、どのようなスタック要素で表現してもよい。ただし、述語 `position ()` を含むときは、スタックが一致した回数を保持するためのカウンタを用意する。

#### 【0049】

図9は、基準点の種類と作成されるスタックとの関係を例示する図である。

また図9において、矢印（直線矢印と波線矢印）は共に、まず元のスタックを比較し、一致すれば先のスタックを比較することを表す。具体的には、直線矢印は、まず元のスタックを比較し、一致したならば、続けて先のスタックを比較することを表す。一方、波線矢印は、まず元のスタックを比較し、一致したならば、その後に特定のXML イベントが通知された場合に限り、先のスタックを比較することを表す。この場合の特定のXML イベントとは、基準点の種類（`following-sibling`か`following`かの別）に応じて異なる。

##### ・ `following-sibling` の場合

元のスタックにおける先頭のスタック要素と一致したノード以降の兄弟ノードとその子孫ノードに対応するXML イベント。兄弟ノードに対応するXML イベントに対応するスタック要素を基準として、先のスタックを比較する。

##### ・ `following` の場合

元のスタックの先頭のスタック要素と一致したノード以降のノードに対応するXML イベント。ただし、その子孫ノードに対応するXML イベントを除く。そのようなノードの各々に対応するXML イベントに対応するスタック要素を基準として、先のスタックを比較する。

#### 【0050】

図10は、図9に示した関係に基づいて、`XPath` から生成されるスタックを示す図である。

- (1) `chapter/para`、
- (2) `chapter/para [@type="warning"]`、
- (3) `chapter/para [2]`、
- (4) `./para`

の各 X P a t h から図 10 に示したスタックがそれぞれ生成される。なお、上記及び図 10 において、c h i l d 基準点は X P a t h の表記規則に基づいて既述が省略されている。

スタック生成部 41 によって、以上のようにして生成されたスタックは、例えば図 1 のメインメモリ 103 の作業領域に保持され、後述する評価実行部 43 の処理において使用される。

なお、本実施の形態では、X P a t h の機能に対して、実施の形態 1 の場合と同様の制限が課される。

#### 【0051】

XML イベント転送部 42 は、XML パーサ 10 から出力された XML イベントを入力し、XML 文書の木構造におけるノードごとに評価実行部 43 へ順次転送する。また、当該 XML イベントをアプリケーション実行部 30 へも順次転送する。

#### 【0052】

評価実行部 43 は、スタック生成部 41 にて生成されたスタックを取得し、XML イベント転送部 42 から転送された XML イベントを受け取って、XML 文書のトークン列を順次読み込みながらスタックを操作する。そして、X P a t h を満たすノードに対応するトークンが読み込まれた時点でその X P a t h が満たされたと評価する。この評価結果は、評価イベントとしてアプリケーション実行部 30 に送られる。

#### 【0053】

本実施の形態では、上記のスタックを用いた X P a t h の評価を実現するためのデータ構造として、実施の形態 1 の場合と同様に、XML 文書の入れ子構造を解析するスタック（XML イベント列用のスタック）を導入する。すなわち、評価実行部 43 は、スタック生成部 41 で作成される上記の X P a t h 用のスタックとは別に、XML イベント列用のスタックを用意し、XML パーサ 10 から送られる XML イベント列を保持する。この XML イベント列用のスタックに関しては、原則として、開始タグに対応する XML イベントを入力したならば、それをスタック要素で表現し、スタックにプッシュする。そして、終了タグに対応す

るXMLイベントを入力したならば、スタックをポップする。また、テキスト等に対応するXMLイベントを入力したならば、このXMLイベントもスタック要素で表現しスタックにプッシュするが、スタックの比較が終了した後、即座に当該スタックをポップする。ここで、XMLイベントは、そのイベントが保持する情報を保持する限り、どのようなスタック要素で表現しても良い。

#### 【0054】

さらに評価実行部43は、XMLイベント転送部42からXMLイベントが転送される度に、XML文書中の着目しているノードのXMLイベントに対応するスタック要素を基準として、XMLイベント列用のスタックとXPath用のスタックとを比較することにより、XPathの評価を行う。ここで、スタックを比較するとは、対応するスタック要素を比較することである。また、スタック要素を比較するとは、対応するXMLイベント（及びそれ以前に通知されたXMLイベント）に対して、対応するノードテストと述語とを評価することである。

#### 【0055】

スタックの比較は、原則として、最後のスタック要素から行う。一致しないスタック要素を検出したならば、直ちに比較処理を終了する。また、毎回全てのスタック要素を比較しても良いし、比較において一致しなかったスタック要素の位置を保持しておき、次の比較処理では当該一致しなかったスタック要素から後を比較するようにしても良い。

スタックが一致したならば、これは、最後に通知されたイベントに対応するノードに対してXPathがマッチしたことを意味する。ただし、XPath用のスタックにカウンタが用意されているときは、スタックが一致する度にXMLイベント列用のスタックにおける対応スタック要素のカウント値をインクリメントし、XPath用のスタックのカウント値と一致したときに限り、XPathがマッチしたと判断する。

#### 【0056】

以上のように、本実施の形態では、評価対象であるXPathをスタック要素の列で表現したXPath用スタック（第1のスタック）を生成し、さらに処理対象であるXML文書を構成するXMLイベント列を入力して入力された個々の

XML イベントに基づいて処理対象の XML 文書の入れ子構造を解析する XML イベント列用スタック（第 2 のスタック）を生成する。そして、第 1 のスタックと当該第 2 のスタックとを比較することにより、X P a t h の部分的な評価を行う。X P a t h の部分的な評価の結果は、XML イベント列用スタックのスタック要素として保持される。

また、上述したように、XML イベント列用スタックの操作は、実施の形態 1 で説明した XML イベント列用スタックの操作と同様である。したがって、この XML イベント列用スタックの操作及び比較が実施の形態 1 で説明したオートマトンの状態遷移に相当している。ただし、実施の形態 2 では、スタックに蓄積されたデータのマッチングのみで X P a t h の評価が実行されるので、実装が容易である。

#### 【0057】

次に、本実施の形態の X P a t h 評価部 40 による処理について、さらに詳細に説明する。

図 11 は、処理対象である XML 文書の例を示す図、図 12 乃至図 14 は、図 11 の XML 文書に対するスタックを用いた評価処理の様子を示す図である。なお、図 11 では、各タグに関して、document を doc とするなど、適宜、省略表記がなされている（以下、他の XML 文書についても同様）。

X P a t h として図 10 の（3）に示した

chapter/para [2]

が与えられたものとする。すなわち、所定の XML 文書のイベント列に対して、ノード chapter の下位に位置する 2 つ目のノード para が検出されたときに、この X P a t h の成立という評価がなされる。この X P a t h に対してスタック生成部 41 にて生成されるスタックは図 10 に示したとおりである。

#### 【0058】

図 12 乃至図 14 には、図 11 の XML 文書に対する SAX イベント列（XML イベント列）と、各 SAX イベント列を入力した際の評価実行部 43 による比

較の様子が順番に示されている。

最初に、ドキュメント開始を示す SAX イベント `startDocument` が入力され、スタックの比較が開始される (図 12 (A))。この時点では、XML イベント列用スタックのスタック要素は空 (NULL) である。次に、要素 (ノード) の開始を示す SAX イベント `startElement: doc` が入力され (図 12 (B))、さらにトークン (SAX イベント) の読み込みが続けられる。

#### 【0059】

次に、要素の開始を示す SAX イベント `startElement: chapter` が入力され、XML イベント列用スタックがプッシュされて XPath 用スタックと比較される (図 12 (C))。このスタック要素は一致するが、この時点ではスタック要素 `para` が一致していないので、トークン (SAX イベント) の読み込みが続けられる。続いて、SAX イベント `startElement: para` が入力され、XML イベント列用スタックがプッシュされて XPath 用スタックと比較される (図 13 (A))。このスタック要素は一致するが、図 11 の XML 文書で最初に現れたノード `para` であるので、ポジション関数 (`pos = 2`) を扱うためのカウンタの値 (`current_pos`) が 1 インクリメントされ、トークン (SAX イベント) の読み込みが続けられる。

#### 【0060】

以後、要素の終了を示す SAX イベント `endElement: para`、`endElement: chapter` が入力されて XML イベント列用スタックの対応するスタック要素がそれぞれポップされ (図 13 (B) (C))、次に再び SAX イベント `startElement: chapter` が入力されて XML イベント列用スタックがプッシュされ、XPath 用スタックと比較される (図 14 (A))。このスタック要素は一致し、続いて SAX イベント `startElement: para` が入力され、XML イベント列用スタックがプッシュされて XPath 用スタックと比較される (図 14 (B))。このスタック要素は一致し、カウンタの値が 1 インクリメントされると、`current_pos = 2` となって、ポジション関数の要求を満たすので、図 11 の XML 文書に対して XPath “`chapter/para[2]`” が成立したと評価される。



## 【0061】

さて、上記の説明で、following-sibling及びfollowingに関しては、ノードはステップKで選択されるノードの候補であると同時にステップK-1で選択されるノードの候補でもあるため、本実施の形態では、ステップK-1が成立したときにプッシュされたノードがポップされた時点でプロセスを2つに分割し、それぞれのプロセスが2通りのロケーションパスの解釈に対応したスタック操作を続行することとした。この動作について、具体例を挙げて詳細に説明する。

図15は、処理対象であるXML文書の例を示す図、図16は評価対象であるXPathとこのXPathから生成されるスタックとを示す図、図17乃至図19は、図15のXML文書に対するスタックを用いた評価処理の様子を示す図である。

## 【0062】

図16に示すように、XPathとして

```
chap/para [@num="2"] /following-sibling::para
```

が与えられたものとする。すなわち、所定のXML文書のイベント列に対して、ノードchapterの下位に位置するノードpara num="2"に後続する兄弟ノードparaが検出されたときに、このXPathの成立という評価がなされる。

## 【0063】

図17乃至図19には、図15のXML文書に対するSAXイベント列（XMLイベント列）と、各SAXイベント列を入力した際の評価実行部43による比較の様子が順番に示されている。

まずドキュメント開始を示すSAXイベントstartDocument、続いて要素の開始を示すSAXイベントstartElement: docが入力され（図17（A）（B））、さらにトークン（SAXイベント）の読み込みが続けられる。次に、要素の開

始を示す SAX イベント `startElement: chapter` が入力され、XML イベント列用スタックがプッシュされて XPath 用スタックと比較されるが (図 17 (C))、スタック要素 `para` が一致していないので、トークン (SAX イベント) の読み込みが続けられる。

#### 【0064】

次に、SAX イベント `startElement: para, @num=1` が入力され、XML イベント列用スタックがプッシュされて XPath 用スタックと比較される (図 18 (A))。このスタック要素は XPath 用スタックの対応するスタック要素 (`para@num=2`) とは一致せず、引き続きトークン (SAX イベント) の読み込みが行われる。続いて、要素の終了を示す SAX イベント `endElement: para` が入力され、XML イベント列用スタックの対応するスタック要素がそれぞれポップされた後 (図 18 (B))、SAX イベント `startElement: para, @num=2` が入力され、XML イベント列用スタックがプッシュされて XPath 用スタックと比較される (図 18 (C))。このスタック要素は XPath 用スタックの対応するスタック要素と一致する。したがって、これ以降は図 16 に示した XPath 用スタックの後半 (スタック要素 `para`) との比較を行う。

#### 【0065】

次に、SAX イベント `endElement: para` が入力され、XML イベント列用スタックの対応するスタック要素がそれぞれポップされた後 (図 19 (A))、SAX イベント `startElement: para, @num=3` が入力され、XML イベント列用スタックがプッシュされて XPath 用スタックと比較される (図 19 (B))。このスタック要素は XPath 用スタックの後半 (スタック要素 `para`) に一致するので、図 15 の XML 文書に対して XPath “`chap/para [@num=“2”]/following-sibling::para`” が成立したと評価される。

#### 【0066】

図 20 は、本実施の形態による XML 文書のストリーミング処理を説明するフローチャートである。

図 6 に示すように、XPath 評価部 40 は、初期動作として、スタック生成

部41により、予め与えられたX P a t hのロケーションパスからX P a t h用スタックを生成しておく（S t e p 2001）。処理対象であるXML文書が入力され、XMLパーサ10からXMLイベントが送られると（S t e p 2002）、X P a t h評価部40の評価実行部43により、上記のようにして、当該XMLイベント用スタックが生成され、X P a t h用スタックとの比較が順次判断される（S t e p 2003）。この比較結果が当該X P a t h用スタックにて表現されたX P a t hの部分的な評価を意味している。

そして、このXMLイベント列用スタックとX P a t h用スタックとが全て一致するまでXMLイベントの入力及びスタックの比較を繰り返し、完全に一致したならば、このXML文書に対してロケーションパスが成立したと評価され、評価イベントがアプリケーション実行部30へ送られる（S t e p 2004、2005）。

アプリケーション実行部30は、XMLパーサ10から出力されたXMLイベントと、X P a t h評価部40の評価実行部43から出力された評価イベントとを取得し、当該評価イベントに応じて当該XMLイベントを順次処理していく。

処理対象のXML文書についてXMLパーサ10から送られる全てのXMLイベントに対して、S t e p 2002～2005の処理を実行した後、本実施の形態によるXML文書のストリーミング処理を終了する（S t e p 2006）。

### 【0067】

#### 〔実施の形態3〕

実施の形態3では、ストリーミング形式でXMLイベント列を入力しながら当該XML文書に対する木（文書木：ドキュメントツリー）を順次作成していき、この文書木の作成に伴って逐次的にX P a t hの評価を行う。

上述したように、実施の形態1では、評価対象であるX P a t hからオートマトンを生成し、ストリーミング形式で入力されるXMLイベント列によって当該オートマトンの状態遷移を順次行うことによってX P a t hの評価を行った。また実施の形態2では、X P a t hのステップを第1のスタックで表現し、ストリーミング形式で入力されるXML文書の入れ子構造を第2のスタックで表現し、これらのスタックの内容を比較することによってX P a t hの評価を行った。こ

これらの手法は、いずれもXMLイベント列のうち評価の済んだ部分については情報を喪失してしまうため、リバース基準点に関しては、非常に限定された条件下でのみ評価が可能であった。

これに対し、本実施の形態では、XMLイベント列の入力に伴って文書木を生成していくので、所定のXMLイベントを入力した際に、それ以前のXMLイベント列に関する情報が全て木構造として保持される。したがって、リバース基準点のステップが含まれるXPathに対して何ら問題なく評価を行うことができる。

#### 【0068】

本実施の形態は、実施の形態1と同様に、例えば図1に示した情報処理装置にて実現される。

図21は、本実施の形態におけるXPath評価手段（解析システム）の実装例を示す図である。

図21に示すように、本実施の形態による情報処理装置は、処理対象であるXML文書の構文解析（パース）を行うXMLパーサ10と、構文解析されたXML文書に対してXPathの評価を行うXPath評価部50と、XPathの評価を経たXML文書を用いて所定の情報処理を実行するアプリケーション実行部30とを備える。

図21に示すXMLパーサ10、XPath評価部50及びアプリケーション実行部30は、例えば、図1に示したメインメモリ103に展開されたプログラムにてCPU101を制御することにより実現される仮想的なソフトウェアブロックである。なお、XMLパーサ10及びアプリケーション実行部30は、実施の形態1における対応する構成要素と同様であるため、同一の符号を付して説明を省略する。

#### 【0069】

XPath評価部50は、XMLパーサ10からXMLイベントを受け取り、予め与えられたXPathの評価を行う。

図22は、XPath評価部50の機能構成を説明する図である。

図22に示すように、本実施の形態のXPath評価部50は、XPath格

納部 51 と、XML イベント転送部 52 と、文書木生成部 53 と、評価実行部 54 とを備える。

#### 【0070】

XPath 格納部 51 は、例えば図 1 に示したコンピュータ装置のメインメモリ 103 で実現され、予め与えられた XPath を保持する。XML イベント転送部 52 は、XML パーサ 10 から出力された XML イベントを入力し、XML 文書の木構造におけるノードごとに文書木生成部 53 へ順次転送する。また、当該 XML イベントをアプリケーション実行部 30 へも順次転送する。

#### 【0071】

文書木生成部 53 は、XML イベント転送部 52 から転送された XML イベントを受け取って、その XML イベントまでの文書木を生成する。この文書木は、取得した XML イベントまでの構造を反映しているので、処理対象である XML 文書全体の構造を表す文書木に対する部分木となる。また、XML イベント列を最後まで取得した際に生成される文書木は、当該 XML 文書全体の文書木と同一となる。

文書木の生成は、XML イベントとして当該 XML 文書の要素に対する開始タグトークンが入力されたときに新たにノードを追加することにより行われる。そして、開始タグトークンの入力によりノードを追加した場合、それ以降の挿入位置を当該ノードとする。また、当該要素に対する終了タグトークンが入力された場合、当該ノードの親ノードをそれ以降の挿入位置とする。したがって、この文書木は、文書木生成部 53 が XML イベントを入力する度に更新されることとなる。

文書木生成部 53 により生成（更新）された文書木は、例えば図 1 のメインメモリ 103 の作業領域に保持され、後述する評価実行部 54 の処理において使用される。

#### 【0072】

評価実行部 54 は、文書木生成部 53 により上記の文書木が更新されると、当該文書木を用いて、XPath 格納部 51 に保持されている XPath の評価を行う。本実施の形態の評価実行部 54 は、XPath を評価して成立したステッ

プに関して、当該ステップにおいて選択された文書木のノードの情報、すなわち  $XPath$  における部分的な評価結果の情報を保存する。この情報は、各ステップのステップIDとノードIDとの組(タプル)の集合  $\{ \langle sid1, nid1 \rangle, \langle sid1, nid2 \rangle, \dots \}$  として表される。これを保存集合と呼ぶ。生成された保存集合は、例えば図1のメインメモリ103の作業領域に保持される。

保存集合において各ステップは、1番から順に番号付けされているとする。ここで、0番目のステップは特別なステップとし、このステップに対応させてルートノードを常に保存しておく。評価実行部54の評価により選択された各ノードは、当該ノードをコンテキストノードとして、それ以降の各ステップによって選択されるノード集合が空とならない限り保存される。

#### 【0073】

図23乃至図25は、評価実行部54による文書木及び保存集合を用いた  $XPath$  の評価手順を示すフローチャートであり、図23、24はXMLイベントとして開始タグトークンが入力された場合の動作、図25はXMLイベントとして終了タグトークンが入力された場合の動作を示す。なお、説明の簡単のため、図示の動作では述語(predicate)を考慮していない。述語への対応については後述する。

図23に示すように、評価実行部54は、XMLイベントとして開始タグトークンが入力されると(Step 2301)、この要素に対応するノードNをそれまでに生成されている文書木に挿入する(Step 2302)。初期的にはルートノードに対してノードNが挿入される。そして、保存集合の各要素  $\langle S, nid \rangle$  に対し、ステップSの次の番号がつけられたステップS'を参照し、以下の動作を行う(Step 2303)。

#### 【0074】

評価実行部54は、まずステップS'がロケーションパス( $XPath$ )の最後かどうかを調べ(Step 2304)、最後ならば、さらにステップS'の評価が成立するかどうかを調べる(Step 2305)。評価が成立したならば、最後に挿入されたノードNが  $XPath$  にてアドレッシングされたノードである

ので、このノードNを評価イベントとして出力する (Step 2306)。

ステップS' がロケーションパスの最後でない場合、次に評価実行部54は、ステップS' の評価が成立するかどうかを調べる (Step 2307)。評価が成立したならば、さらにステップS' の次のステップS'' における基準点に応じて、次の処理を行う。

なお、上記のStep 2305、2307の判断で、ステップS' の評価が成立しない場合、当該XMLイベントに対する処理を終了して、次のXMLイベントの入力を待つ。

#### 【0075】

図24に示すように、ステップS'' の基準点がself基準点である場合、ステップS'' を新たにステップS' として、評価実行部54の処理がStep 2304の判断へ戻る (Step 2308、2309)。

#### 【0076】

ステップS'' の基準点がchild、descendant等のフォワード基準点である場合、まず、ノードNをコンテキストノードとして、ステップS'' の基準点が指定する関係を満たす入力があるかどうかを判定する (Step 2308、2310)。そのような入力の可能性があるならば、タプル (ステップIDとノードIDとの組) <S', N>を保存集合に追加する (Step 2311)。そして、既に構築された文書木におけるノードNに対してドキュメント順で後に位置するノード集合 (selfを含む基準点 (descendant-or-self等) の場合はノードN自身を含む) の各要素を新しいノードNとし、ステップS'' を新たにステップS' として、評価実行部54の処理がStep 2304の判断へ戻る (Step 2312)。

ステップS'' の基準点が指定する関係を満たす入力の可能性がないならば、タプル<S', N>を保存集合に追加することなく、当該ノードNに対してドキュメント順で後に位置するノード集合 (selfを含む基準点 (descendant-or-self等) の場合はノードN自身を含む) の各要素を新しいノードNとし、ステップS'' を新たにステップS' として、評価実行部54の処理がStep 2304の判断へ戻る (Step 2310、2312)。

## 【0077】

ステップS”の基準点がparent、ancestor等のリバース基準点である場合、既に構築された文書木におけるノードNに対してドキュメント順で後に位置するノード集合（selfを含む基準点（ancestor-or-self等）の場合はノードN自身を含む）の各要素を新しいノードNとし、ステップS”を新たにステップS’として、評価実行部54の処理がStep 2304の判断へ戻る（Step 2308、2312）。

なお、以上の説明では、評価実行部54はXPathが成立した場合に評価イベントを出力する（ステップ2306参照）こととしたが、XMLイベントを1つ処理する度に成立または不成立を示す評価イベントを出力するようにしても良いことは、実施の形態1、2の場合と同様である。

## 【0078】

一方、図25に示すように、評価実行部54は、XMLイベントとして終了タグトークンが入力されると（Step 2501）、そのトークンに対応したノードを含むタプル<S, N>を保存集合の中から検出する（Step 2502）。次に、検出されたタプルに対して、ノードNをコンテキストノードとして、ステップSの次の番号が付されたステップS’の基準点が指定する関係を満たす入力がある、この後にあり得るかどうかを判定する（Step 2503）。そして、そのような入力の可能性のないタプルを保存集合から除去する（Step 2504）。

この後、処理対象であるトークンの子ノードを含むタプル<S’, N’>を保存集合の中から検出する（Step 2505）。次に、検出されたタプルに対して、ノードN’をコンテキストノードとして、ステップS’の次の番号が付されたステップS”の基準点が指定する関係を満たす入力がある、この後にあり得るかどうかを判定する（Step 2506）。そして、そのような入力の可能性のないタプルを同様に保存集合から除去する（Step 2507）。

## 【0079】

次に、上記の動作説明で省いた述語への対応について説明する。上述した動作におけるノードの評価において、述語を評価するために必要な条件がそろってい



る場合は、述語の評価を行う。述語の評価を行っていない場合は、述語が未評価であることを意味する付加情報をノードや保存集合に付しておき、述語の評価が仮に成功したとみなして、上記の動作を継続する。ただし、述語が未評価のノードに依存するノードの出力は保留する。述語が未評価の保存集合内の要素に対する評価は、その後の入力によって述語の評価条件がそろった時点で行う。述語にロケーションパスが使われている場合は、評価の目的を述語を評価することに変更して図23乃至図25の処理を適用する。

最終的に述語の評価が成功した場合は、その評価の成功に依存していたノードを評価イベントとして出力する。一方、評価が失敗した場合は、保存集合から当該ノードを取り除き、さらに当該ノードの評価の成功を仮定して行われた全ての操作を取り消す。

#### 【0080】

次に、具体例を挙げて本実施の形態によるXPathの評価を説明する。

図26は、処理対象であるXML文書、そのSAXイベント列および文書木を示す図である。

XPath評価部50に、次のXPath

```
/a/b/preceding-sibling::c/following-sibling::d
```

が与えられたものとする。すなわち、ノードaの子供であるノードbに先行する兄弟ノードcに後続する兄弟ノードdが検出されたときに、このXPathの成立という評価がなされる。

#### 【0081】

図27乃至図36には、図26のXML文書に対するSAXイベント列と、これに対応する文書木、及び当該文書木を用いたXPathの評価によって得られる保存集合の遷移の様子が順番に示されている。

最初に、ドキュメント開始を示すSAXイベントstartDocumentが入力され、対応する文書木としてルートノードdocが生成される(図27)。また、タブ

ル (ステップIDとノードIDとの組)  $< 0, doc >$  が保存集合に入れられる。続いて、要素 (ノード) の開始を示す SAX イベント `startElement: "a"` が入力され、ルートノード `doc` に子ノード "a" が付加される (図 28)。また、XPath のステップ 1 までが成立するので、タプル  $< 1, "a" >$  が保存集合に入れられる。

#### 【0082】

次に、SAX イベント `startElement: "c"` が入力され、ノード "a" に子ノード "c" が付加され (図 29)、続いて、要素の終了を示す SAX イベント `endElement: "c"` が入力される (図 30)。次に、SAX イベント `startElement: "b"` が入力されると、ノード "a" に子ノード "b" が付加される (図 31)。このとき、ノード "b" の生成により XPath のステップ 2 が成立すると共に、リバース基準点 (`preceding-sibling`) のステップ 3 も直ちに成立する。したがって、ステップ 2 に関するタプルは保存集合に保持されず、ステップ 3 に関するタプル  $< 3, "c" >$  が保存集合に入れられる。

#### 【0083】

この後、SAX イベント `endElement: "b"` が入力され (図 32)、引き続き、SAX イベント `startElement: "d"` が入力されると、ノード "a" に子ノード "d" が付加される (図 33)。このとき、ノード "d" の生成により XPath のステップ 4 (最後のステップ) まで成立するので、当該 XPath が成立したと評価される。したがって、XPath 評価部 50 の評価実行部 54 からアプリケーション実行部 30 へ、評価イベントとしてノード "d" が出力される (図 23、Step 2306 参照)。

#### 【0084】

この後、SAX イベント `endElement: "d"` が入力され (図 34)、続いて SAX イベント `endElement: "a"` が入力されて、ノード "a" に新たな子ノードが付加される可能性がなくなったので、保存集合からステップ 1 のタプル  $< 1, "a" >$  が除去される (図 35、図 25 の Step 2503、2504 参照)。そしてさらに、ドキュメント終了を示す SAX イベント `endDocument` が入力されて、処理対象である XML 文書のイベント列の入力が終了する (図 36)。

**【0085】**

図37は、本実施の形態によるXML文書のストリーミング処理を説明するフローチャートである。

図37に示すように、XPath評価部50は、処理対象であるXML文書が入力され、XMLパーサ10からXMLイベントが送られると（Step 3701）、まず取得したXMLイベントに基づいて文書木が生成される（Step 3702）。この文書木は、XMLイベントが順次入力されるに伴って更新され、最終的に処理対象のXML文書の文書構造に対応する文書木となる。そして、XPath評価部50の評価実行部54により、この文書木を用いてXPathの評価が行われ、部分的に成立した箇所（ステップ）に関する情報が必要に応じて保存集合に蓄積される（Step 3703）。そして、XPath全体が成立するまでXMLイベントの入力、文書木の生成およびXPathの評価が繰り返され、XPath全体が成立すると評価されたならば、評価イベントがアプリケーション実行部30へ送られる（Step 3704、3705）。

アプリケーション実行部30は、XMLパーサ10から出力されたXMLイベントと、XPath評価部50の評価実行部54から出力された評価イベントとを取得し、当該評価イベントに応じて当該XMLイベントを順次処理していく。

処理対象のXML文書についてXMLパーサ10から送られる全てのXMLイベントに対して、Step 3701～3705の処理を実行した後、本実施の形態によるXML文書のストリーミング処理を終了する（Step 3706）。

**【0086】**

以上、3つの実施の形態によって、本発明を説明した。これらの実施の形態は、ストリーミング形式で入力されるXMLイベント列に対してXPathの部分的な評価を順次行っていき、かかる部分的な評価に関する情報を蓄積していった、最終的にXPath全体の評価結果を得る。

実施の形態1、2は、XMLイベント列のストリーミング形式による入力に応じて部分的に評価していくことが可能なデータ構造にXPathを変換しておき、個々のXMLイベントの入力に応じて順次評価を行った。一方、実施の形態3は、XMLイベント列をストリーミング形式で入力しながらXML文書の文書木

を作成していき、この文書木を用いて逐次的に XPath の評価を行った。

#### 【0087】

実施の形態 1、2 は、評価の済んだ XML イベントに関する情報が失われるので、リバース基準点のステップに関しては、極めて限定された条件下以外では評価できない。これに対し、実施の形態 3 は、評価を行った XML イベントの情報は、文書木として残っているので、リバース基準点のステップに関する評価に対する制限はない。

#### 【0088】

また、実施の形態 1、2、3 のいずれも、全ての XML イベント列を入力していない段階で XPath が成立すると判断された場合には、その段階でアプリケーション実行部 30 による処理を開始することが可能である（ストリーミング処理）。しかしながら、実施の形態 3 は、XPath がリバース基準点のステップを含む場合に、XML イベントの通知と評価イベントのそれに時間差が生じることがあり、また評価イベントの通知が必ずしもドキュメント順（ドキュメントの XML 表現において各ノードの XML 表現の最初の文字が現われる順序）で行われるとは限らないため、アプリケーション実行部 30 による処理が評価イベントによって XML イベントの処理を切り替えるようなものである場合は注意が必要である。実施の形態 1、2 にはかかる制限はない。

#### 【0089】

したがって、XPath がリバース基準点のステップを含まない場合には、実施の形態 1、2 を用いる。そして、XPath がリバース基準点のステップを含む場合であっても、アプリケーション実行部 30 による処理が評価イベントによって XML イベントの処理を切り替えるようなものでない場合や、そのようなものであっても処理の遅延が許されるような場合には、実施の形態 3 を用いるといった使い分けが可能である。

実施の形態 2 は、入力した XML イベント列をスタックに列の情報のまま格納し、XPath を表現するスタックの記憶内容と比較するだけなので、メモリの使用量が非常に少なく、かつ処理に要する時間も短い。また、比較的容易に実装可能である。したがって、いずれの実施の形態も実施可能である場合、実施の形

態 2 を用いることが好ましい。

なお、本発明の技術的範囲は、上記 3 つの実施の形態に限るものではなく、技術的思想において同一である種々の変形を含むことは言うまでもない。

【0090】

【発明の効果】

以上説明したように、本発明によれば、XML 文書をストリーミング処理しながら XPath の評価を行う解析システム及びその解析方法を実現することができる。

【図面の簡単な説明】

【図 1】 実施の形態 1 を実現するのに好適な情報処理装置のハードウェア構成の例を模式的に示した図である。

【図 2】 実施の形態 1 における XPath 評価手段（解析システム）の実装例を示す図である。

【図 3】 実施の形態 1 における XPath 評価部の機能構成を説明する図である。

【図 4】 実施の形態 1 におけるオートマトン生成部にて生成されるオートマトンを模式的に示す図である。

【図 5】 実施の形態 1 による処理に用いられるロケーションパス、オートマトン及び処理対象の XML 文書の例を示す図である。

【図 6】 実施の形態 1 による XML 文書のストリーミング処理を説明するフローチャートである。

【図 7】 実施の形態 2 における XPath 評価手段（解析システム）の実装例を示す図である。

【図 8】 実施の形態 2 における XPath 評価部の機能構成を説明する図である。

【図 9】 基準点の種類と本実施の形態にて作成されるスタックとの関係を例示する図である。

【図 10】 図 9 に示した関係に基づいて、XPath から生成されるスタックを示す図である。

【図 1 1】 処理対象である XML 文書の例を示す図である。

【図 1 2】 図 1 1 の XML 文書に対するスタックを用いた評価処理の様子を示す図である。

【図 1 3】 図 1 1 の XML 文書に対するスタックを用いた評価処理の様子を示す図である。

【図 1 4】 図 1 1 の XML 文書に対するスタックを用いた評価処理の様子を示す図である。

【図 1 5】 処理対象である XML 文書の他の例を示す図である。

【図 1 6】 評価対象である X P a t h とこの X P a t h から生成されるスタックとを示す図である。

【図 1 7】 図 1 5 の XML 文書に対するスタックを用いた評価処理の様子を示す図である。

【図 1 8】 図 1 5 の XML 文書に対するスタックを用いた評価処理の様子を示す図である。

【図 1 9】 図 1 5 の XML 文書に対するスタックを用いた評価処理の様子を示す図である。

【図 2 0】 実施の形態 2 による XML 文書のストリーミング処理を説明するフローチャートである。

【図 2 1】 実施の形態 3 における X P a t h 評価手段（解析システム）の実装例を示す図である。

【図 2 2】 実施の形態 3 における X P a t h 評価部の機能構成を説明する図である。

【図 2 3】 実施の形態 3 の評価実行部による文書木及び保存集合を用いた X P a t h の評価手順を示すフローチャートであり、XML イベントとして開始タグトークンが入力された場合の動作を示す図である。

【図 2 4】 実施の形態 3 の評価実行部による文書木及び保存集合を用いた X P a t h の評価手順を示すフローチャートであり、XML イベントとして開始タグトークンが入力された場合の動作を示す図である。

【図 2 5】 実施の形態 3 の評価実行部による文書木及び保存集合を用いた

XPathの評価手順を示すフローチャートであり、XML イベントとして終了タグトークンが入力された場合の動作を示す図である。

【図 26】 処理対象であるXML 文書、そのSAX イベント列および本実施の形態で生成される文書木を示す図である。

【図 27】 XML 文書に対するSAX イベント列、文書木、保存集合の遷移の様子を示す図であり、SAX イベントstartDocumentが入力されたときの様子を示す図である。

【図 28】 XML 文書に対するSAX イベント列、文書木、保存集合の遷移の様子を示す図であり、SAX イベントstartElement: "a"が入力されたときの様子を示す図である。

【図 29】 XML 文書に対するSAX イベント列、文書木、保存集合の遷移の様子を示す図であり、SAX イベントstartElement: "c"が入力されたときの様子を示す図である。

【図 30】 XML 文書に対するSAX イベント列、文書木、保存集合の遷移の様子を示す図であり、SAX イベントendElement: "c"が入力されたときの様子を示す図である。

【図 31】 XML 文書に対するSAX イベント列、文書木、保存集合の遷移の様子を示す図であり、SAX イベントstartElement: "b"が入力されたときの様子を示す図である。

【図 32】 XML 文書に対するSAX イベント列、文書木、保存集合の遷移の様子を示す図であり、SAX イベントendElement: "b"が入力されたときの様子を示す図である。

【図 33】 XML 文書に対するSAX イベント列、文書木、保存集合の遷移の様子を示す図であり、SAX イベントstartElement: "d"が入力されたときの様子を示す図である。

【図 34】 XML 文書に対するSAX イベント列、文書木、保存集合の遷移の様子を示す図であり、SAX イベントendElement: "d"が入力されたときの様子を示す図である。

【図 35】 XML 文書に対するSAX イベント列、文書木、保存集合の遷

移の様子を示す図であり、SAX イベント endElement: "a" が入力されたときの様子  
を示す図である。

【図 36】 XML 文書に対する SAX イベント列、文書木、保存集合の遷移の様子を示す図であり、SAX イベント endDocument が入力されたときの様子  
を示す図である。

【図 37】 実施の形態 3 による XML 文書のストリーミング処理を説明する  
フローチャートである。

【符号の説明】

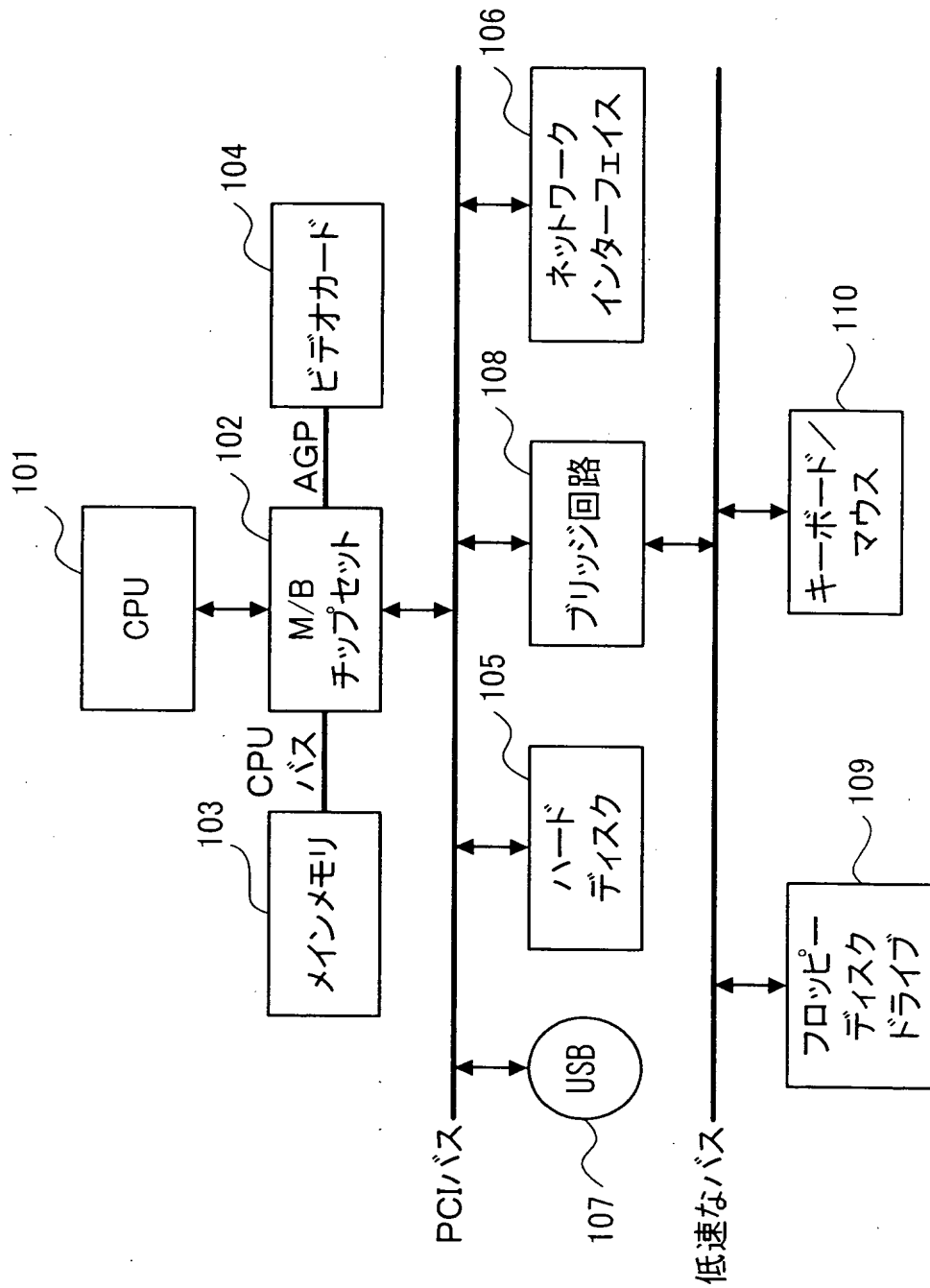
10…XML パーサ、20、40、50…XPath 評価部、21…オートマトン生成部、22、42、52…XML イベント転送部、23、43、54…評価実行部、30…アプリケーション実行部、41…スタック生成部、51…XPath 格納部、53…文書木生成部、101…CPU、103…メインメモリ



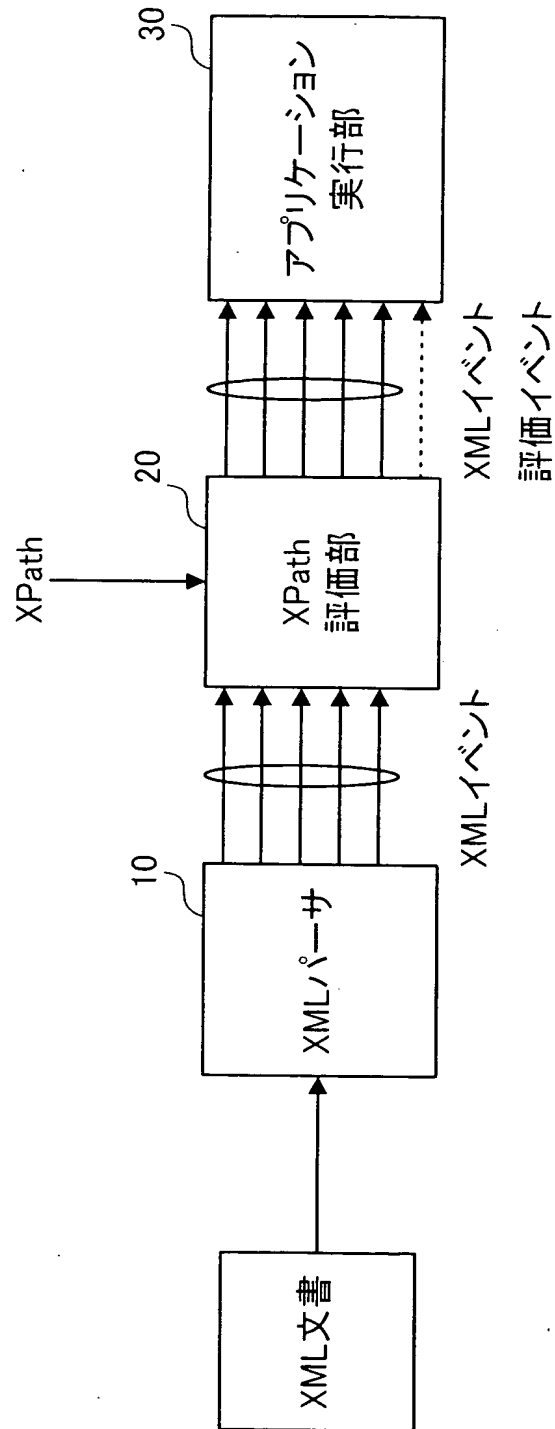
【書類名】

図面

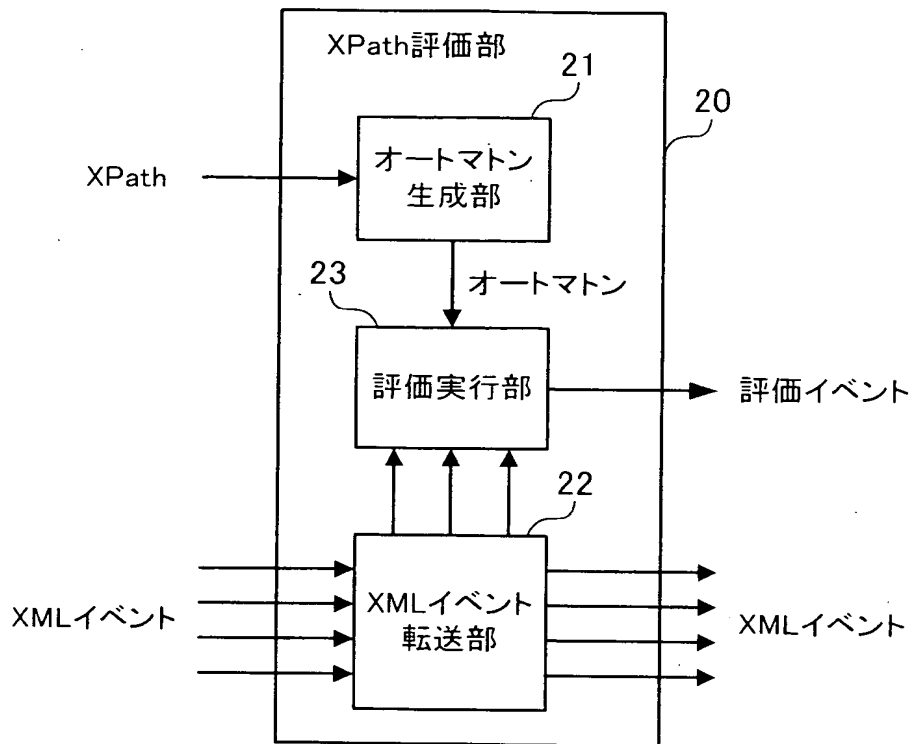
【図 1】



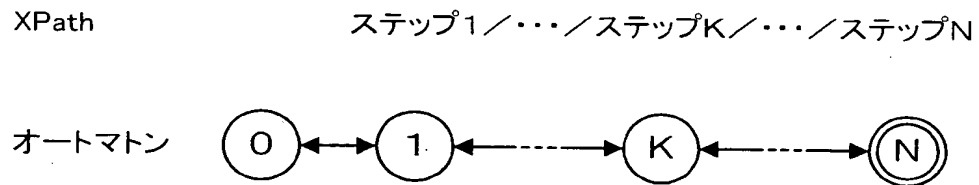
【図 2】



【図 3】



【図 4】



【図 5】

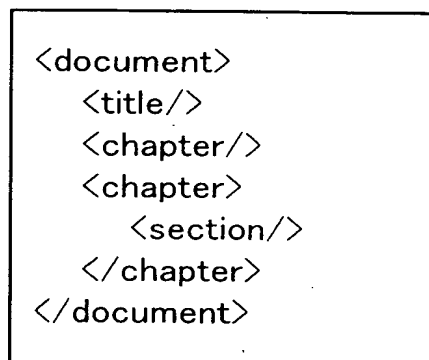
(A)

ステップ1      ステップ2      ステップ3  
/child::document/child::chapter/child::section

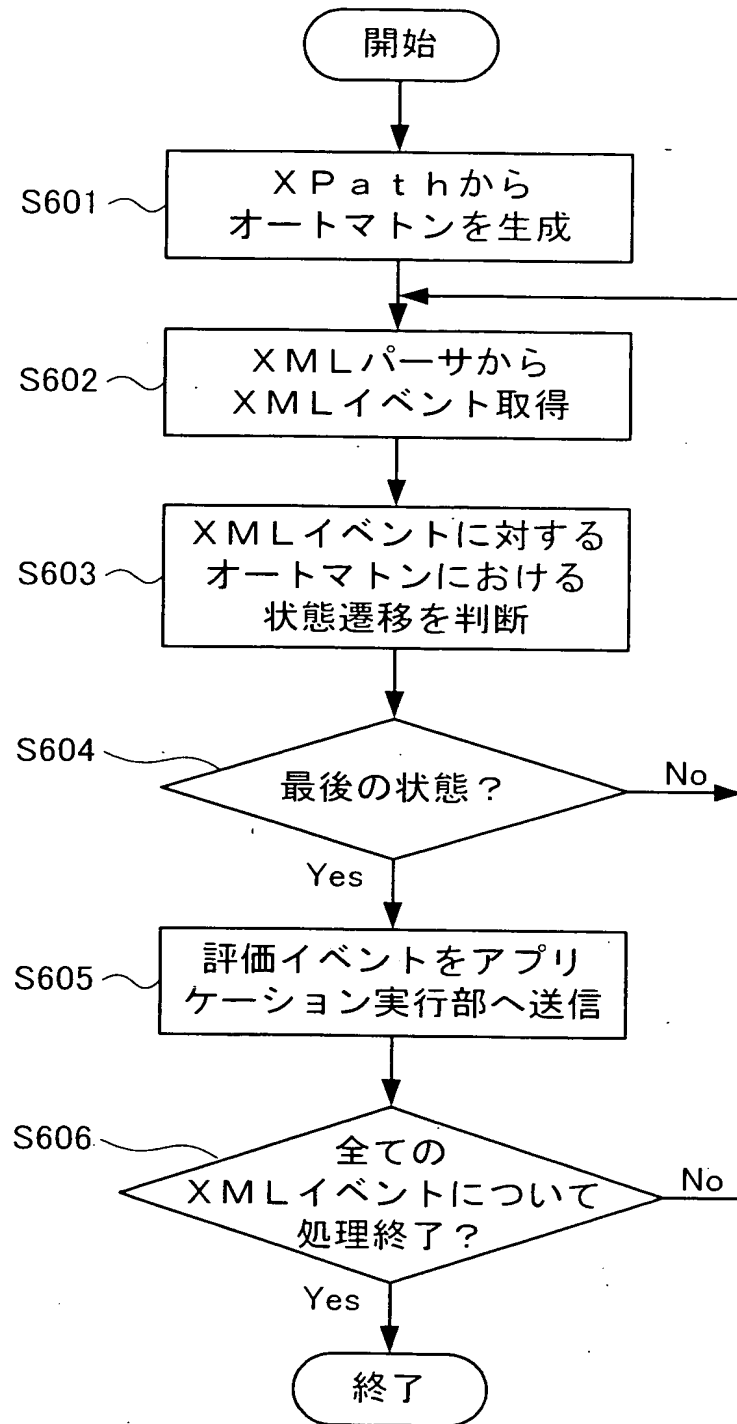
(B)



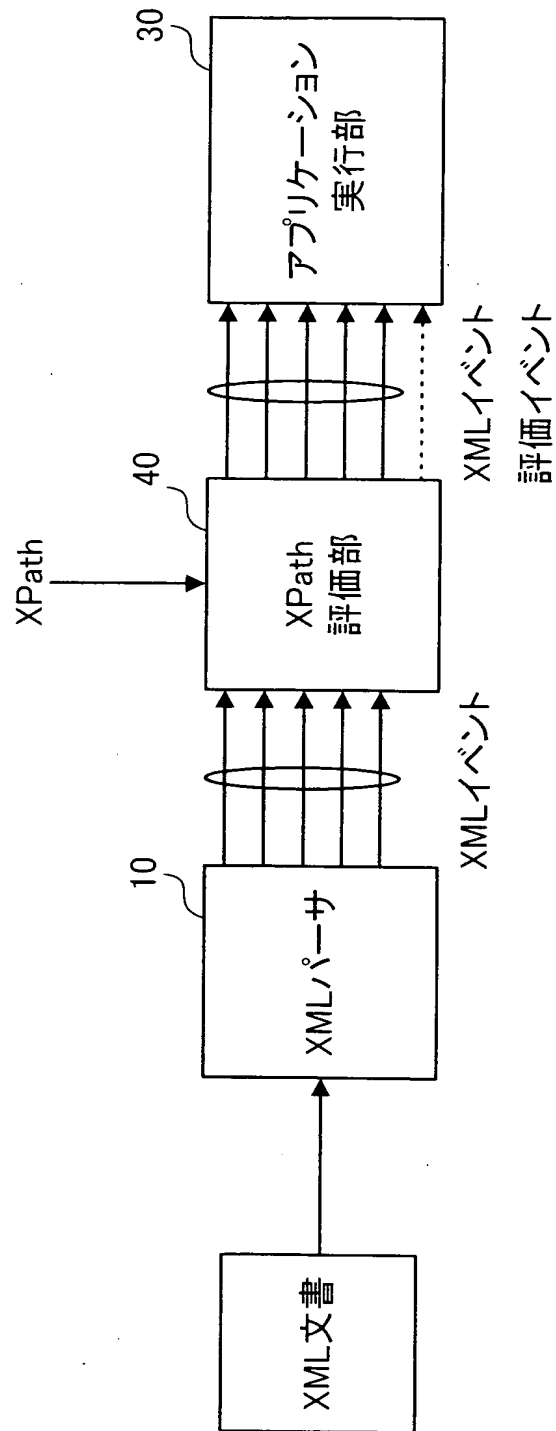
(C)



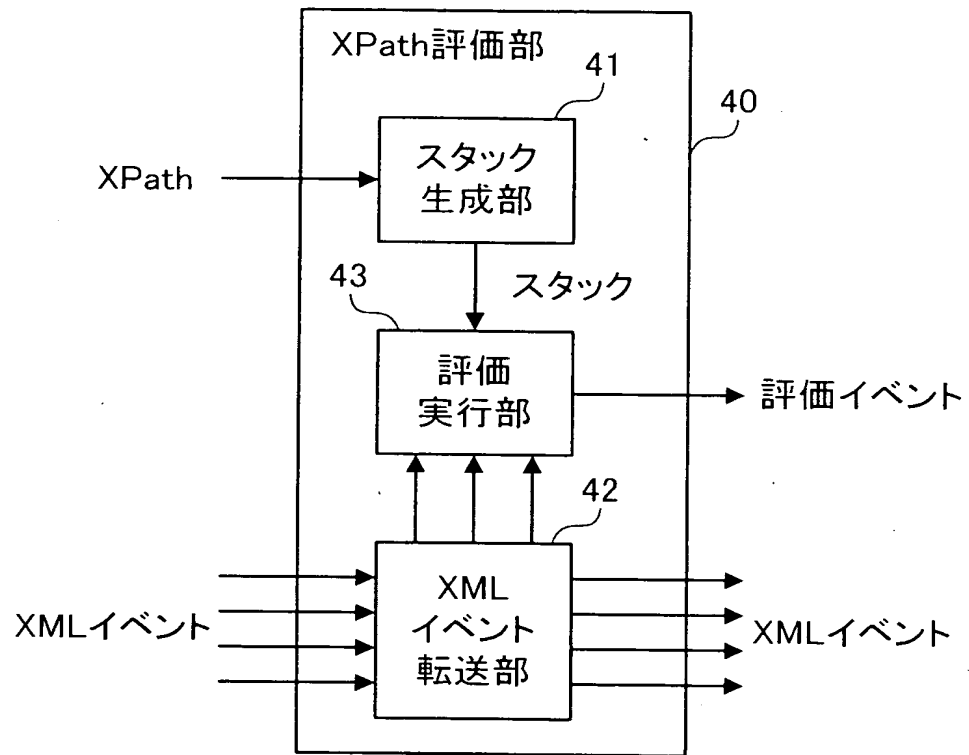
【図 6】



【図 7】

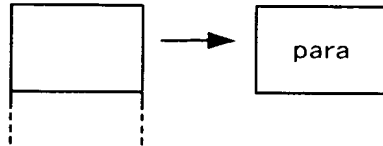


【図 8】

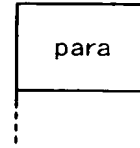


【図 9】

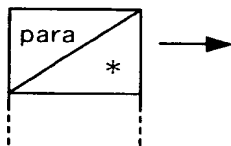
▪ self



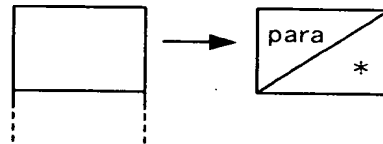
▪ child



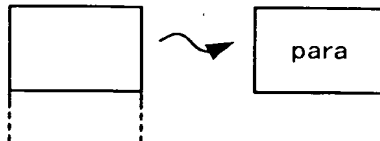
▪ descendant



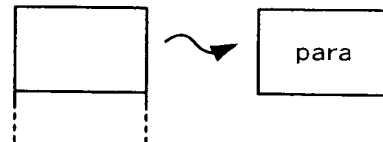
▪ descendant-or-self



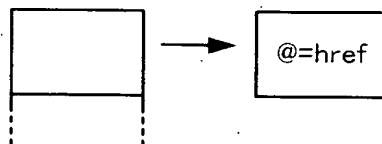
▪ following



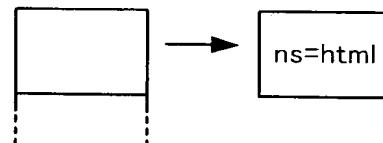
▪ following-sibling



▪ attribute



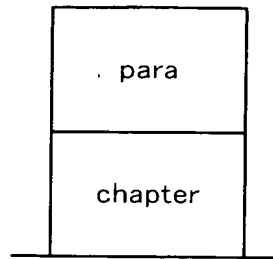
▪ namespace



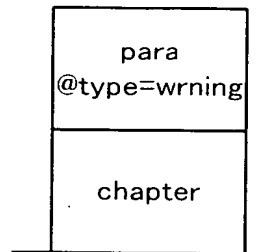


【図 10】

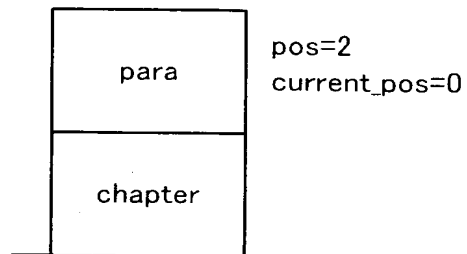
(1) chapter/para



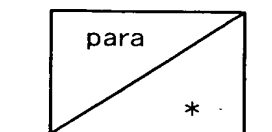
(2) chapter/para[@type="warning"]



(3) chapter/para[2]



(4) ././para



【図 1 1】

```
<doc>
  <chapter>
    <para>xxx</para>
  </chapter>
  <chapter>
    <para>yyy</para>
  </chapter>
</doc>
```

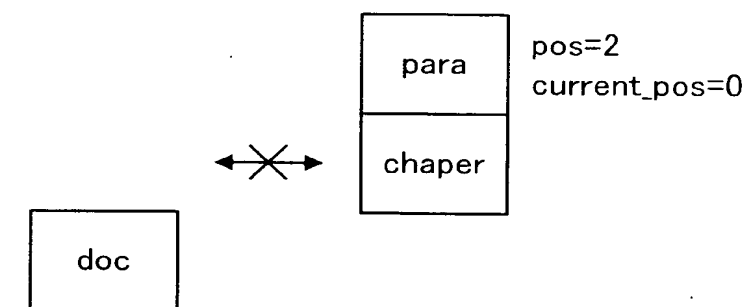
【図 12】

(A) startDocument

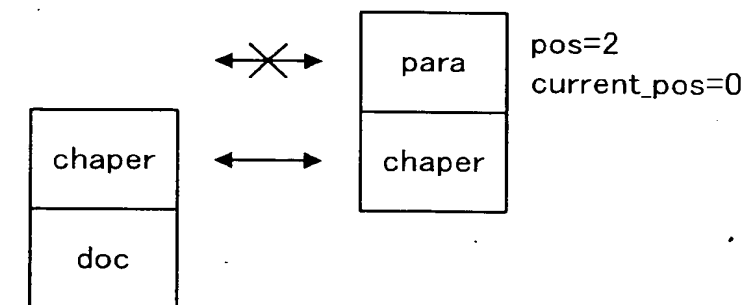
XMLイベント列用  
スタック

XPath用スタック

(B) startElement: doc

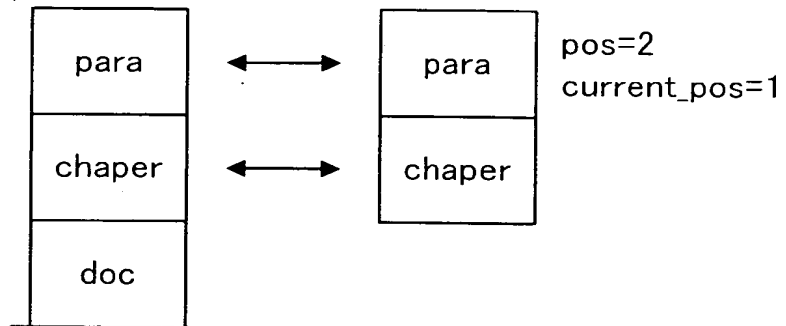


(C) startElement: chapter

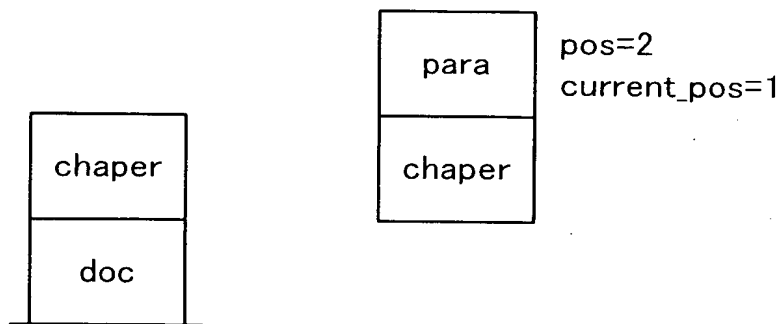


【図 13】

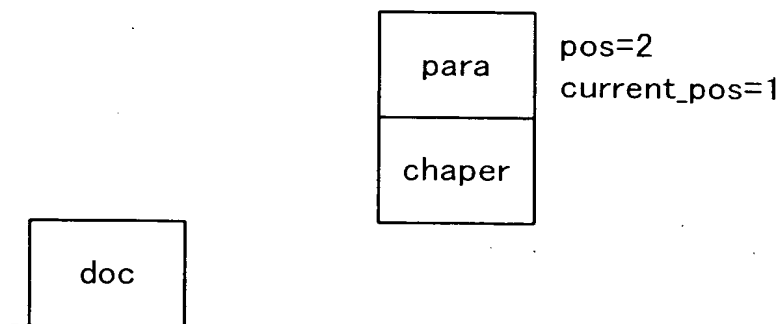
(A) startElement: para



(B) endElement: para

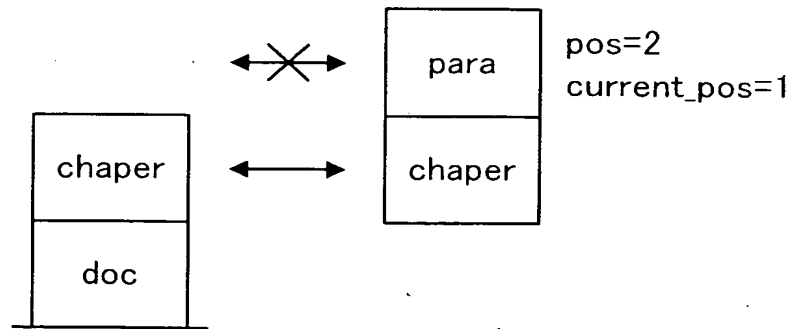


(C) endElement: chapter

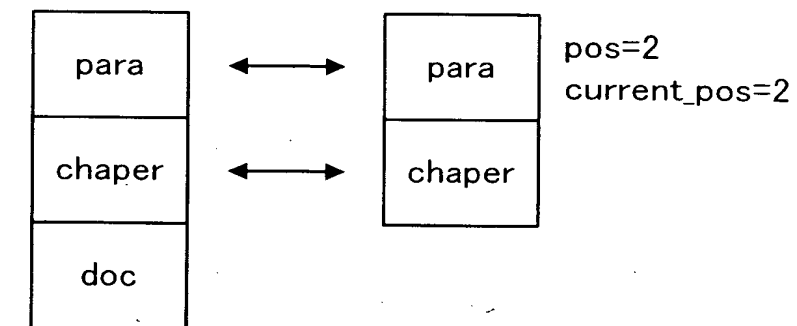


【図 14】

(A) startElement: chapter



(B) startElement: para



【図 15】

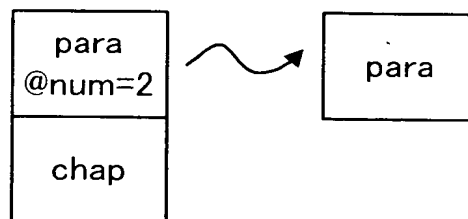
```
<doc>
  <chap>
    <para num="1"/>
    <para num="2"/>
    <para num="3"/>
  </chap>
  <chap/>
</doc>
```

【図 16】

XPath      chap/para[@num="2"]/following-sibling::para

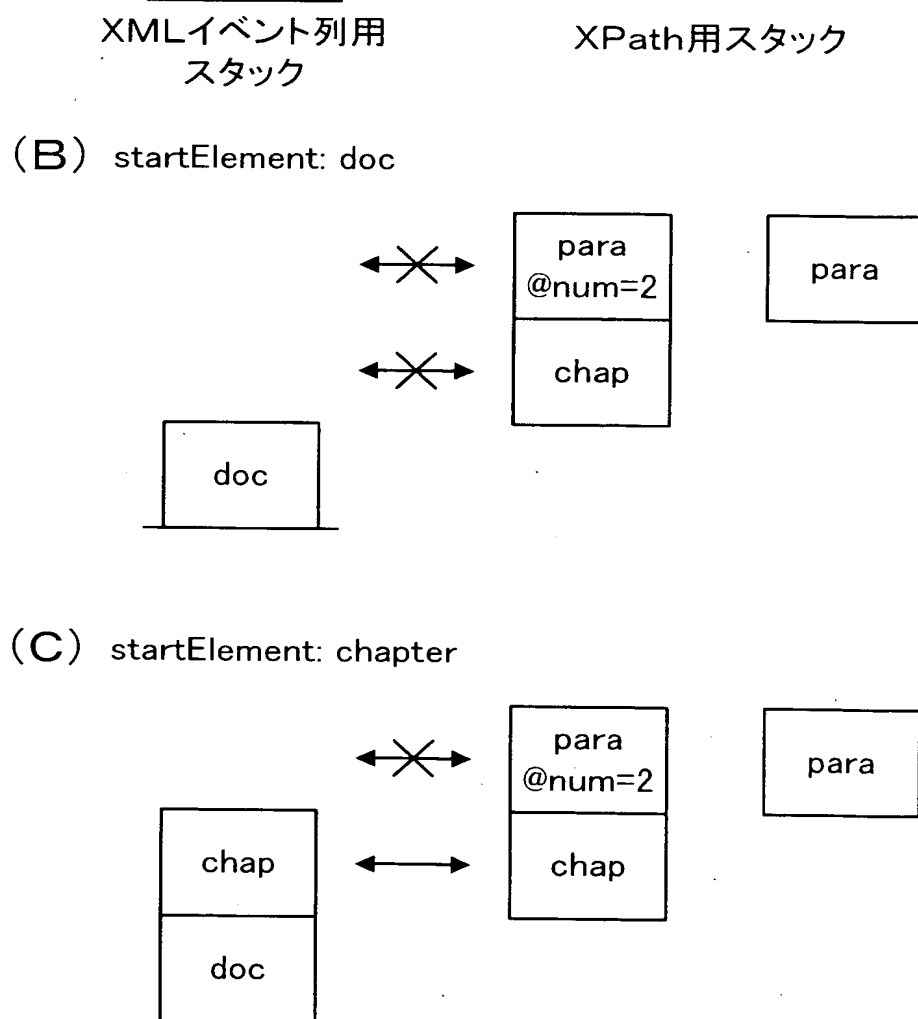


スタック



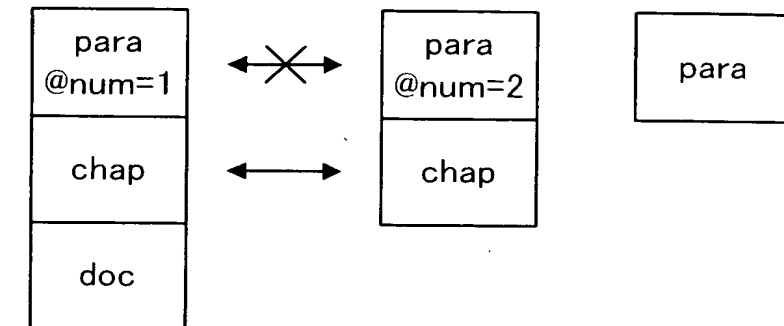
【図 17】

(A) startDocument

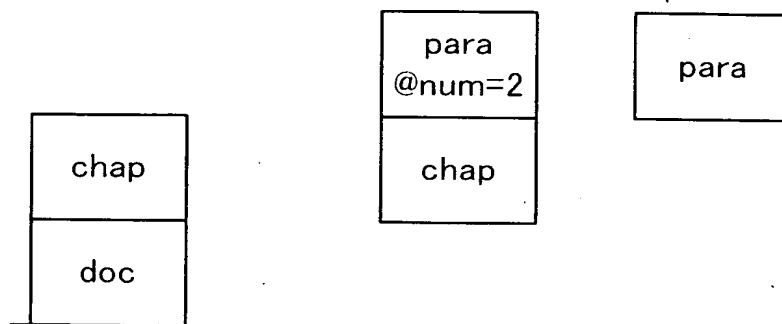


【図 1 8】

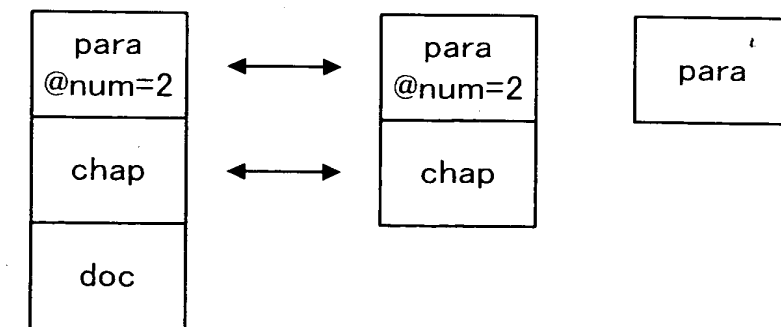
(A) startElement: para, @num=1



(B) endElement: para



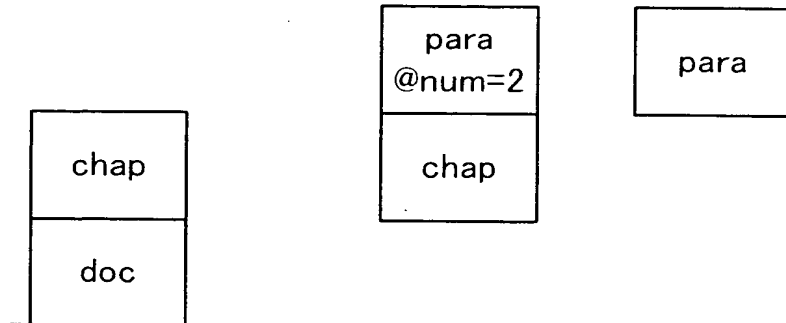
(C) startElement: para, @num=2



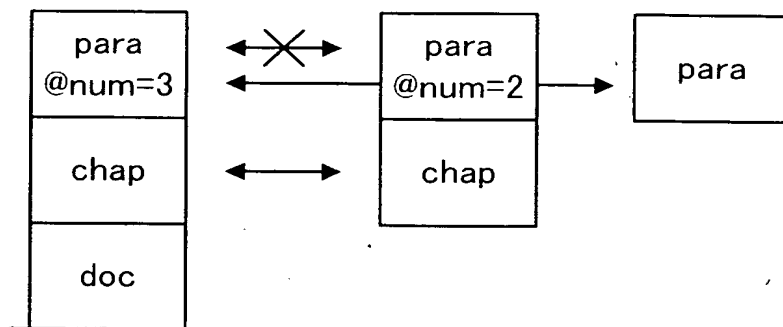


【図 19】

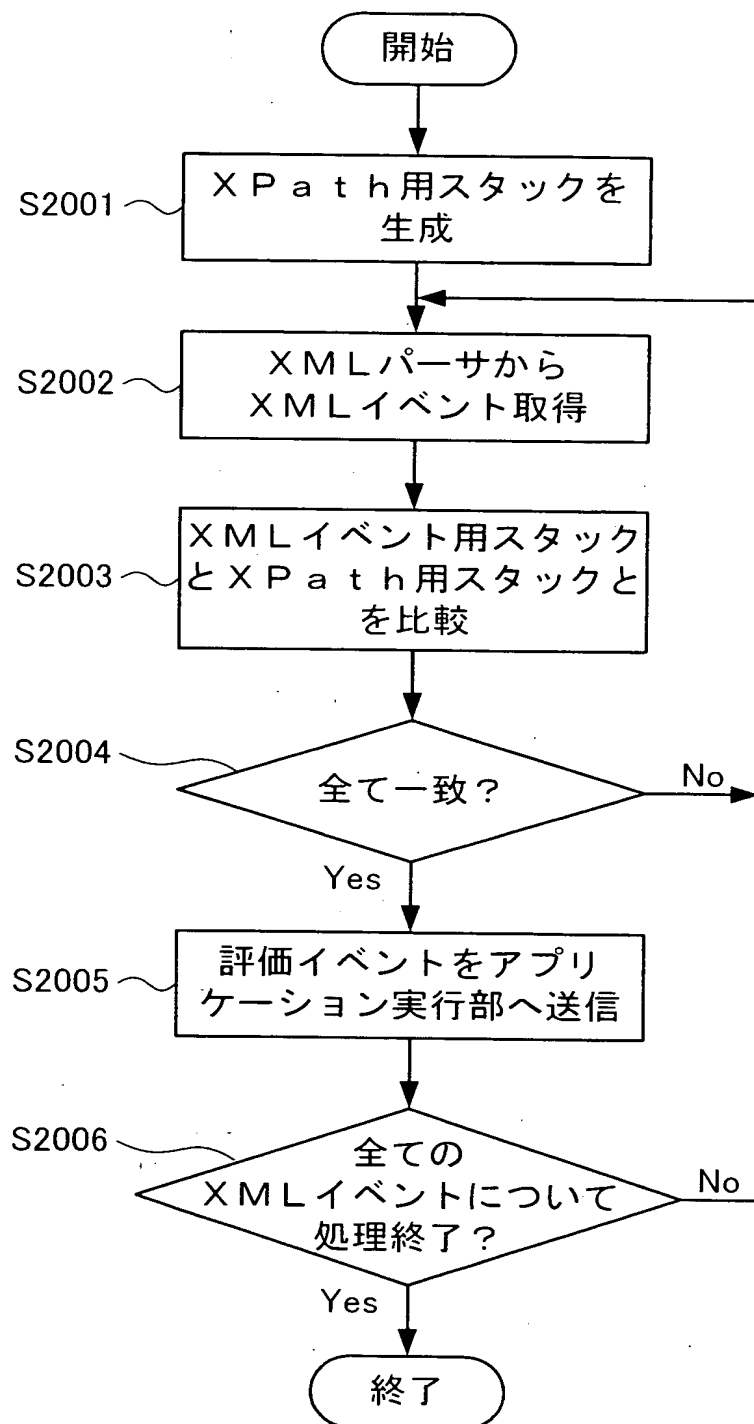
(A) endElement: para



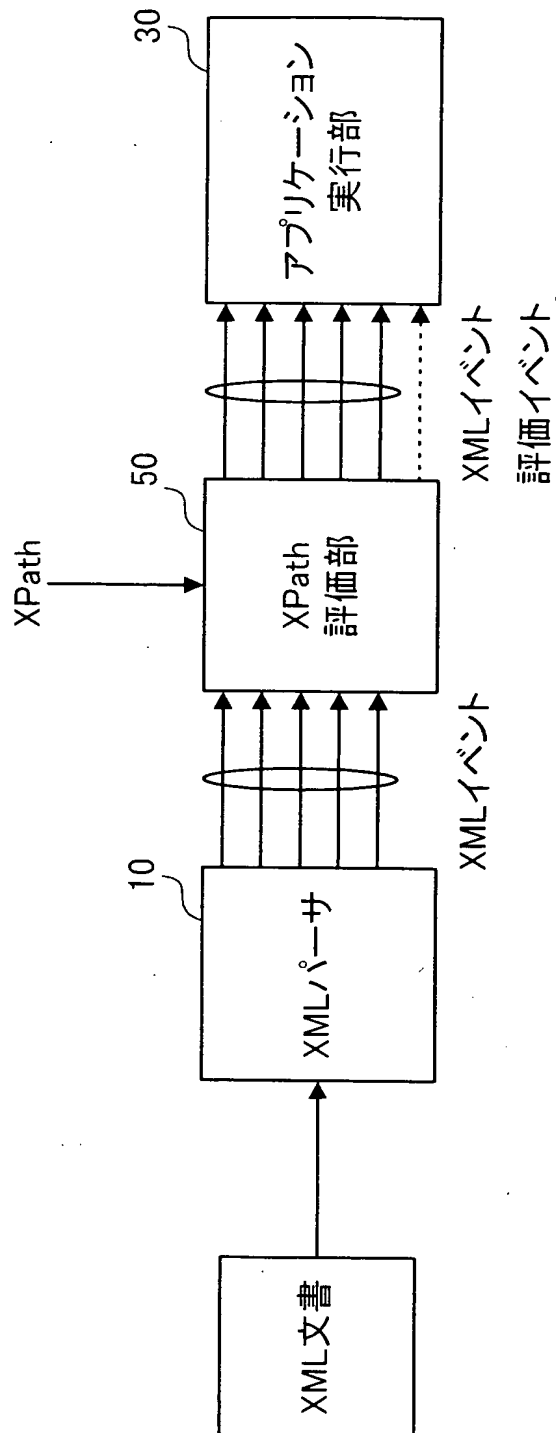
(B) startElement: para, @num=3



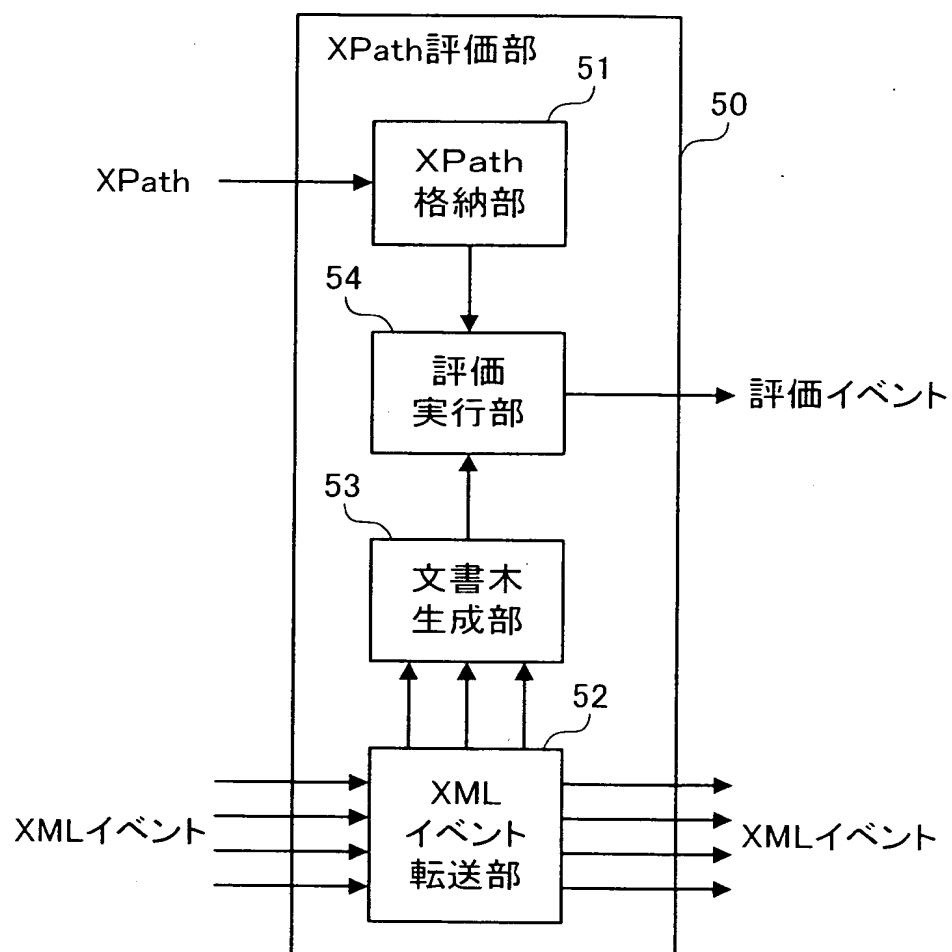
【図 20】



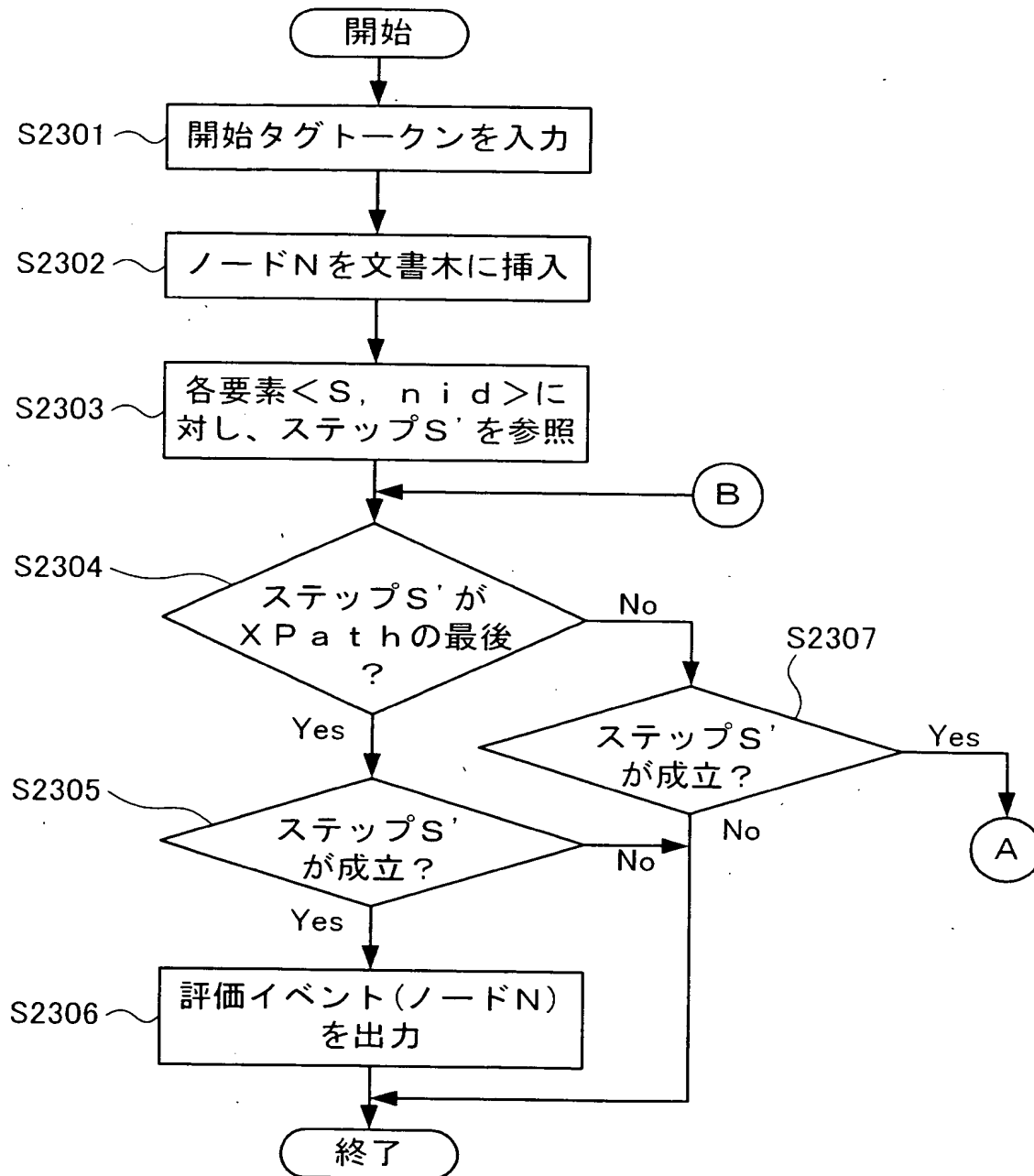
【図 21】



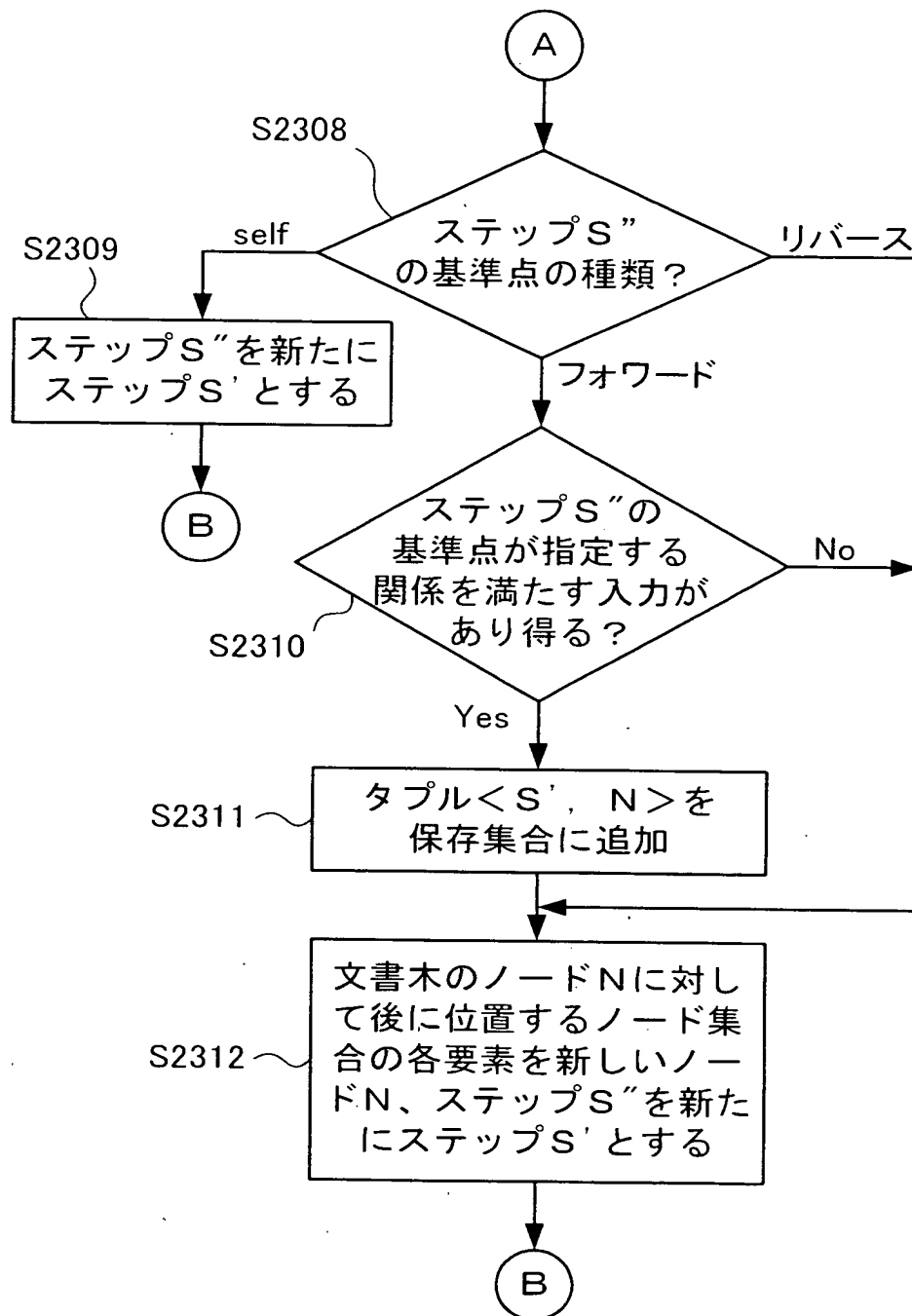
【図 22】



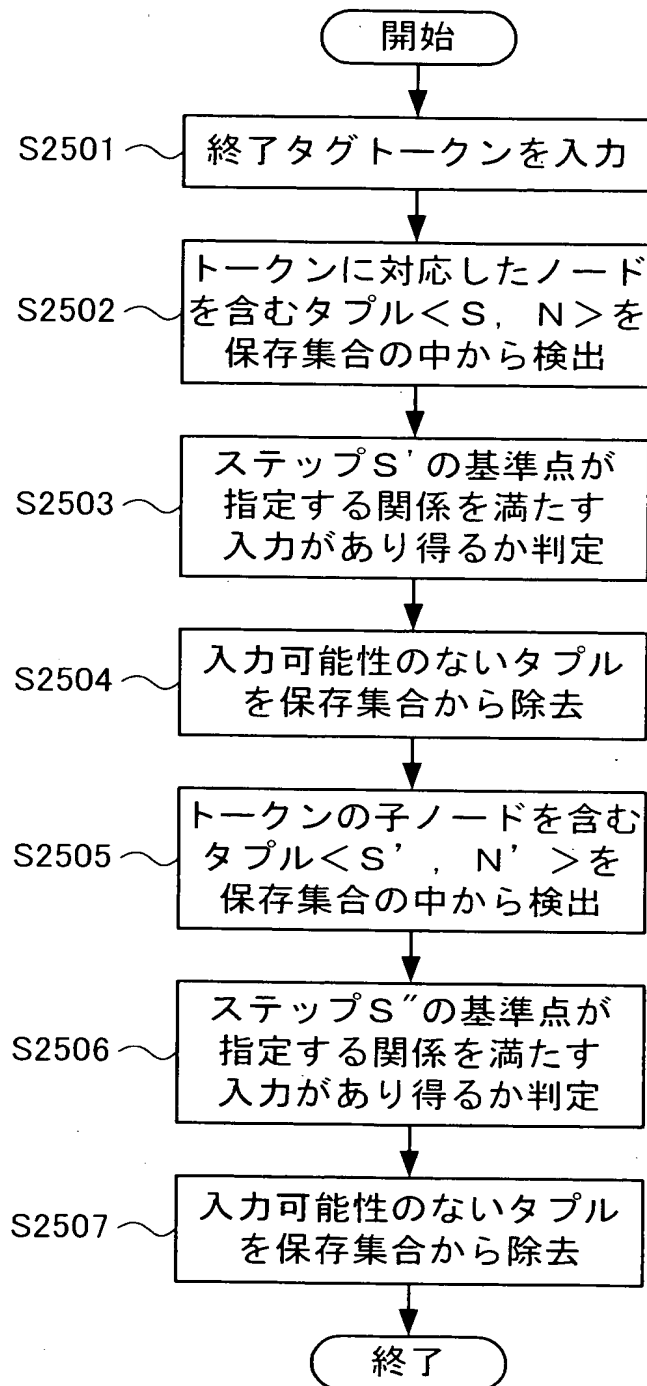
【図 23】



【図 24】



【図 25】



【図 26】

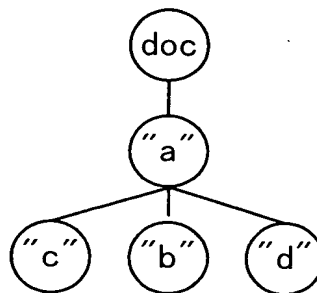
XML 文書

```
<a>  
  <c/>  
  <b/>  
  <d/>  
</a>
```

SAX イベント列

```
startDocument  
startElement:"a"  
startElement:"c"  
endElement:"c"  
startElement:"b"  
endElement:"b"  
startElement:"d"  
endElement:"d"  
endElement:"a"  
endDocument
```

文書木





【図 27】

1	2	3	4
"/a/b/preceding-sibling::c/	following-sibling::d"		

startDocument

doc

{<0, doc>}

SAXイベント列

文書木

保存集合

【図 28】

1	2	3	4
"/a/b/preceding-sibling::c/	following-sibling::d"		

startDocument  
startElement:"a"

doc  
"a"

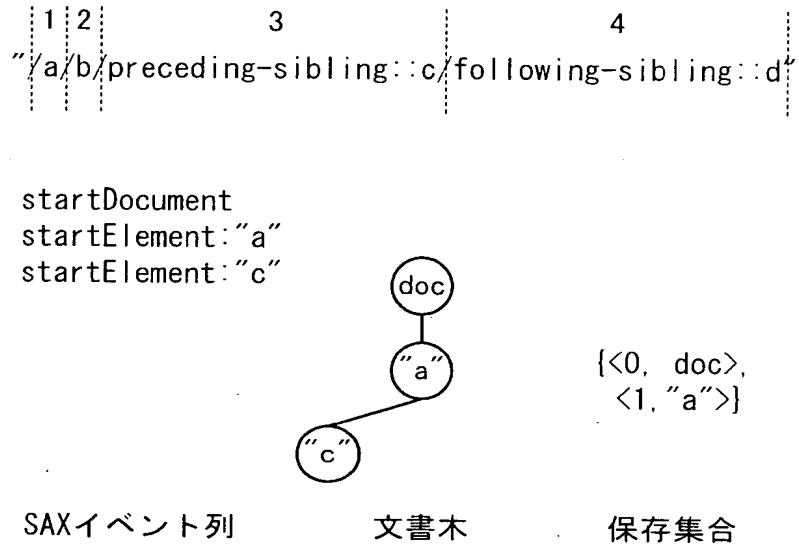
{<0, doc>,  
<1, "a">}

SAXイベント列

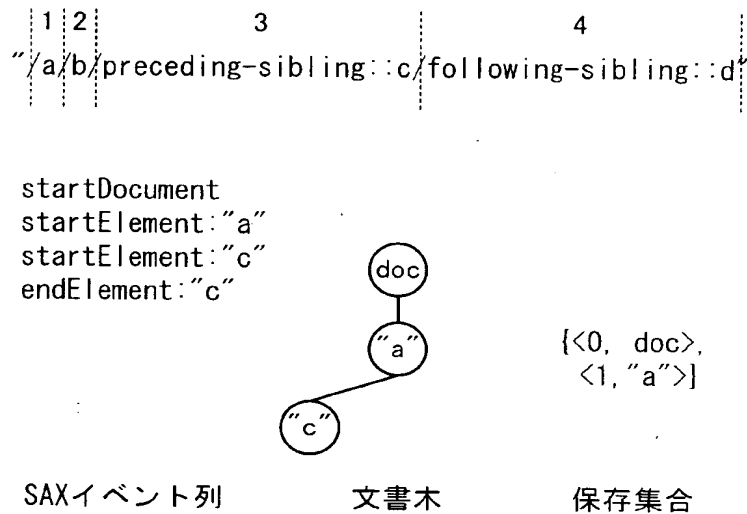
文書木

保存集合

【図 29】



【図 30】



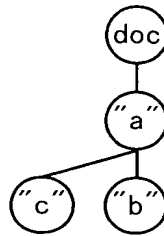
【図 3 1】

1	2	3	4
"/a/b/preceding-sibling::c/following-sibling::d"			

```

startDocument
startElement:"a"
startElement:"c"
endElement:"c"
startElement:"b"

```



```

{<0, doc>,
<1, "a">,
<3, "c"> }

```

SAXイベント列

文書木

保存集合

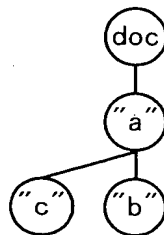
【図 3 2】

1	2	3	4
"/a/b/preceding-sibling::c/following-sibling::d"			

```

startDocument
startElement:"a"
startElement:"c"
endElement:"c"
startElement:"b"
endElement:"b"

```



```

{<0, doc>,
<1, "a">,
<3, "c"> }

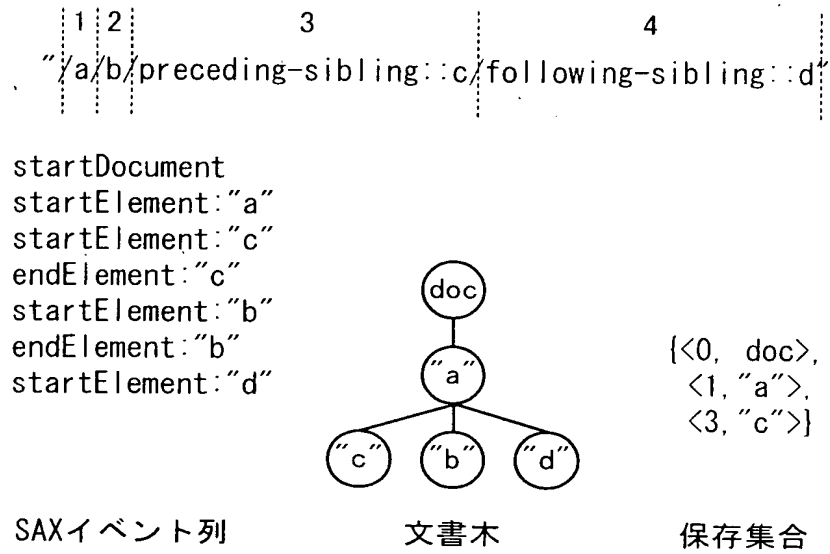
```

SAXイベント列

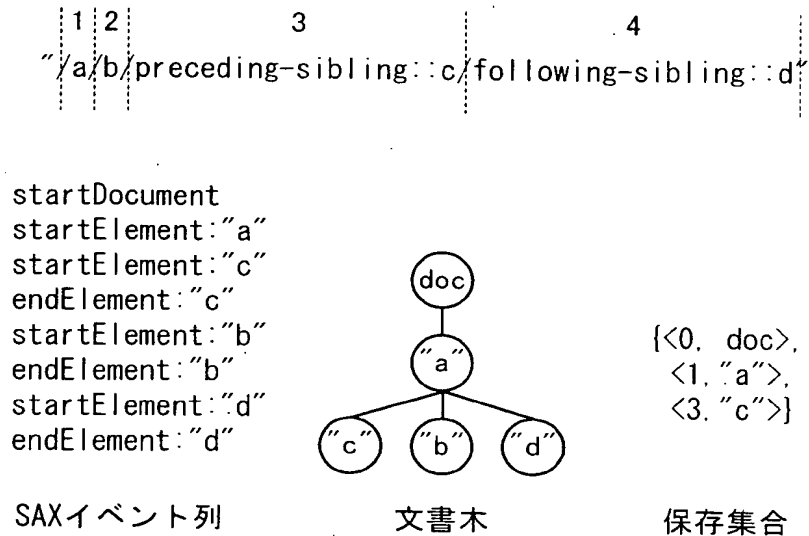
文書木

保存集合

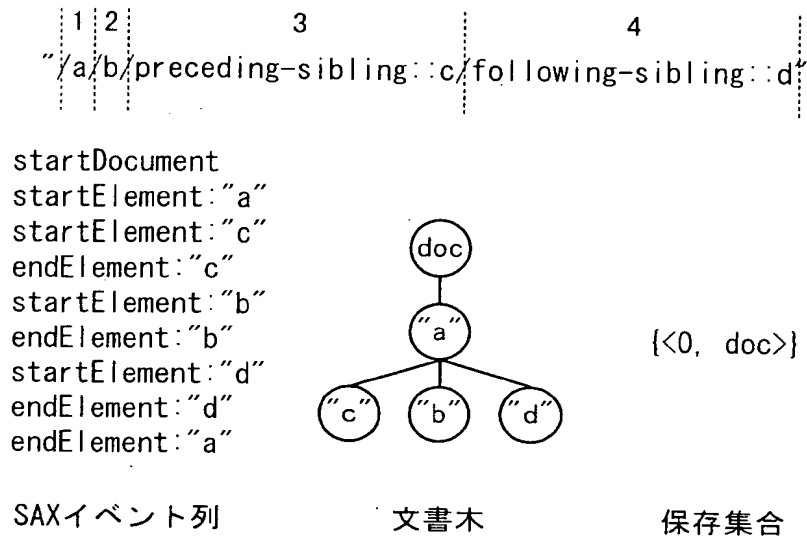
【図 3 3】



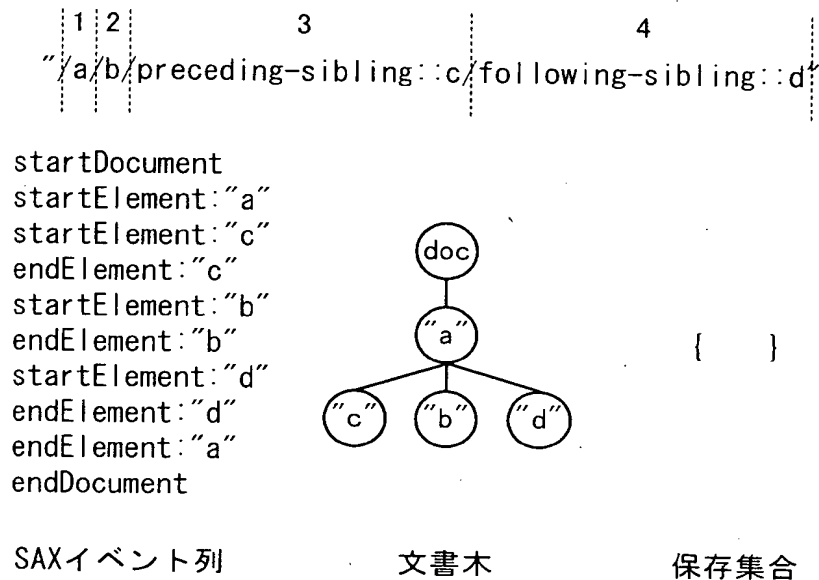
【図 3 4】



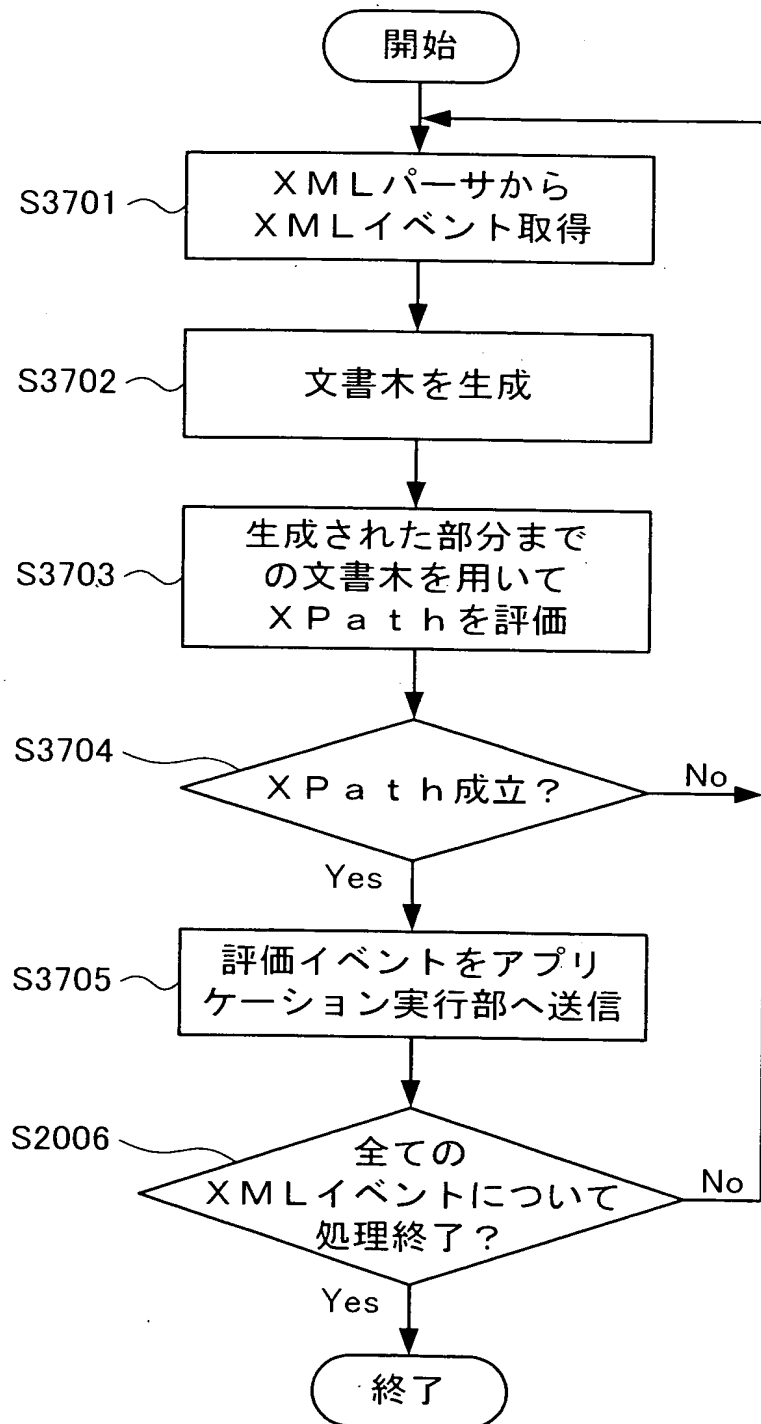
【図 3 5】



【図 3 6】



【図 37】



【書類名】 要約書

【要約】

【課題】 XML文書をストリーミング処理しながらXPathの評価を行う解析システム及びその解析方法を実現する。

【解決手段】 処理対象のXML文書を構成するXMLイベント列を入力するXMLパーサ10と、ストリーミング処理によるXPathの評価を実行するXPath評価部20とを備える。このXPath評価部20は、XMLパーサ10から転送された個々のXMLイベントに対してXPathを順次評価すると共に、所定のXMLイベントに対して、このXPathが部分的に成立した場合に、このXPathの部分的な評価の結果に関する情報を保持していく。そして、このXPathの最後のステップが成立した場合に、XML文書に対して、このXPathが成立したと評価する。

【選択図】 図2

## 認定・付加情報

特許出願の番号 特願 2003-122292  
受付番号 50300703710  
書類名 特許願  
担当官 小野寺 光子 1721  
作成日 平成15年 4月30日

## &lt;認定情報・付加情報&gt;

## 【特許出願人】

【識別番号】 390009531  
【住所又は居所】 アメリカ合衆国10504、ニューヨーク州 アーモンク ニュー オーチャード ロード  
【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレーション

## 【代理人】

【識別番号】 100086243  
【住所又は居所】 神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内  
【氏名又は名称】 坂口 博

## 【代理人】

【識別番号】 100091568  
【住所又は居所】 神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内  
【氏名又は名称】 市位 嘉宏

## 【代理人】

【識別番号】 100108501  
【住所又は居所】 神奈川県大和市下鶴間1623番14 日本アイ・ビー・エム株式会社 知的所有権  
【氏名又は名称】 上野 剛史

## 【復代理人】

申請人  
【識別番号】 100104880  
【住所又は居所】 東京都港区赤坂5-4-11 山口建設第2ビル 6F セリオ国際特許事務所  
【氏名又は名称】 古部 次郎

## 【選任した復代理人】

【識別番号】 100118201

次頁有



認定・付加情報（続き）

【住所又は居所】 東京都港区赤坂 5-4-11 山口建設第二ビル  
6 F セリオ国際特許事務所  
【氏名又は名称】 千田 武

次頁無

特願 2003-122292

出 願 人 履 歴 情 報

識別番号

[390009531]

1. 変更年月日 2000年 5月16日  
[変更理由] 名称変更  
住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)  
氏 名 インターナショナル・ビジネス・マシーンズ・コーポレーション
2. 変更年月日 2002年 6月 3日  
[変更理由] 住所変更  
住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク ニュー オーチャード ロード  
氏 名 インターナショナル・ビジネス・マシーンズ・コーポレーション